CSE 575: Statistical Machine Learning Assignment #2

Instructor: Prof. Jingrui He
Out: Feb 18, 2018; Due: March 18, 2018

Submit electronically, using the submission link on Blackboard for Assignment #1, 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 [15 points]

SVM is an algorithm which fits a *the max-margin hyperplane* in a certain feature space (depending on which kernel is used).

1.1 **(5 points) Hard-margin Linear SVM.** Given a linearly separable data set where each example is either from class $y_i = 1$ or class $y_i = -1$, then a linear SVM can always find the max-margin classifier that correctly classifies all the training data as follows.

$$\arg\max_{\mathbf{w},b} d = d^{+} - d^{-} = \frac{\mathbf{w}^T\mathbf{x}_j}{||\mathbf{w}||_2} - \frac{\mathbf{w}^T\mathbf{x}_k}{||\mathbf{w}||_2} \tag{1}$$

Suppose that we know there are only two training examples on the margins, i.e., the support vectors, $\mathbf{x}_j, y_j = 1$ and $\mathbf{x}_k, y_k = -1$, and the parameters for the linear SVM are $\mathbf{w}*$ and $b*$. Please write down the constraints this linear SVM has to satisfy with $\mathbf{x}_j, \mathbf{x}_k, \mathbf{w}*, b*$, and form the problem of linear SVM as a constrained optimization problem (**Hint: The optimization problem should use both d in Equation (1) as well as the constraints we just worked out.**).

1.2 **(10 points) Soft-margin SVM.** However, in real-world cases, it is difficult to confirm if a dataset is separable or not, even in kernel space. Therefore, soft-margin SVM is introduced to handle those examples which could hardly be correctly classified.
(a). Please describe how the incorrectly classified training examples are handled in a soft-margin SVM.
(b). After training a soft-margin SVM, we can have three types (not classes) of examples based on the value of the slack variables. Let's use ξ_i to denote the value of slack variable for training example $\mathbf{x}_i$. Please define the three types of examples using ξ_i. For each type, if a training example is removed fromt the training set, would the decision boundary change? Please justify your answer.

2 AdaBoost [25 points]

For 2.1 and 2.2, you are given a **balanced** binary training data set, i.e., the number of training examples labeled +1 equals the number of training examples labeled -1.

2.1 **(5 points)** Adaboost is an ensemble of weak classifiers whose accuracy is slightly over 50%. What will happen to your AdaBoost algorithm if your weak classifier has exactly 50% accuracy (a.k.a a pure random classifier)?

2.2 **(5 points)** And what will happen if you use a weak binary classifier whose classification accuracy is **less than** 50%, say 45%?

For 2.3 to 2.5, you are given another simple training data set (different from the one used in 2.1 and 2.2) shown below, which consists of 10 samples (x, y) in Table 1. Here y is the desired class label for each corresponding x.

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

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

2.3 **(5 points)** In the first iteration, we let the threshold θ=2.5 which minimizes the classification error at the current iteration. Show us how you derive the weights D_2 after this iteration.

2.4 **(5 points)** Show us how you derive D_3 and D_4 after the second and third iterations, respectively.

2.5 **(5 points)** Show us what happens at the fourth iteration (**Hint: This part will be easy if you have done all above correctly**).

3 K-Nearest Neighbor Classifier [60 points]

3.1 **(10 points) A Lazy Classifier.** Considering the online learning setting where besides the observed training data, we have new data becoming available as time goes by, some classifiers have to be re-trained from scratch. (**Hint: For some classifiers, there have been extensive work on extending them to the online learning setting efficiently. However, for this question, if the parameters in the classifier need to be adjusted with new data, we consider the classifier being re-trained from scratch.**)
(a). When a new training example becomes available, among SVM, Naive Bayes and KNN, which classifier(s) have to be re-trained from scratch? Please justify your answer.
(b). When a new test example becomes available, among SVM, Naive Bayes and KNN, which classifier needs the most computation to infer the class label for this example, and what is the time complexity for this inference, assuming that we have n training examples, and the number of features is significiantly smaller than n? Please justify your answer.

3.2 **(50 points) Implementation of KNN Classifier.** Evaluate the KNN classifier implemented by you on the famous MNIST data set where each example is a hand written digit. Each example includes 28x28 grey-scale pixel values as features and a categorical class label out of 0-9. You can manually download the dataset from Dr. Yann Lecun's webpage (http://yann.lecun.com/exdb/mnist/) or automatically get it from some libraries/packages (e.g., as done in section 5.9 of http://scikit-learn.org/stable/datasets/index.html for sklearn in Python).

Like assignment #1, you have to implement a KNN classifier with euclidean distance from scratch, **do not use existing class or functions (e.g., sklearn.neighbors.KNeighborsClassifier)**. On the original data set, please use the first 6,000 examples for training, and the last 1,000 examples for testing.

(a). Briefly describe how you implement the KNN classifier by giving the pseudocode. The pseudocode must include equations for how the distances are computed and how classification is done for each example in the test phase. 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 yourFirstName-yourLastName.zip IN ADDITION TO the pdf file (or .doc/.docx file).

(b). Plot curves for training and test errors: the training/test error (which is equal to 1.0-accuracy) vs. the value of K. Plot 11 points for the curve, using $K = 1, 9, 19, 29, 39, 49, 59, 69, 79, 89, 99$. Plot the error curves for training error and test error in the same figure.