CSE6740 - Computational Data Analysis

Homework 2

The TA Team

Georgia Institute of Technology

October 2, 2023

Question 1(a)

Since the data are not linearly separable, we do not expect the algorithm to converge. Describe how you choose your stopping criterion.

Question 1(a)

Since the data are not linearly separable, we do not expect the algorithm to converge. Describe how you choose your stopping criterion.

Hints: Checkout the scikit-learn repo page for the Linear Perceptron: `https://github.com/scikit-learn/scikit-learn/blob/c69cabda0/sklearn/linear_model/_perceptron.py#L9`

Question 1(b)

Once you have implemented your algorithm, you may check your accuracy against a standard implementation such as sklearn's Perceptron function. In particular, check your implementation's accuracy against the classification accuracy (0-1 loss) of sklearn's implementation on the same test data (from above). Explain any hyperparameter choices you make in calling sklearn's Perceptron, and whether these cause a discrepancy in the accuracy.

Question 1(c)

Consider the data $X = \left[x_1, \cdots, x_m\right]$, where $x_i \sim D$ were iid. For this part alone, assume that, with probability 1, the data are linearly separable with margin ρ and that $|x| \leq R$. Prove that the Perceptron converges in at most $\left(\frac{R}{\rho}\right)^2$ steps.

Question 1(c)

Consider the data $X = \left[x_1, \cdots, x_m\right]^T$, where $x_i \sim D$ were iid. For this part alone, assume that, with probability 1, the data are linearly separable with margin ρ and that $|x| \leq R$. Prove that the Perceptron converges in at most $\left(\frac{R}{\rho}\right)^2$ steps.

Hints:

1. Recall Perceptron convergence proof in Lecture 6.
2. Recall Definition of margin in Lecture 7.

Question 1(d)

Give your reasoning for how you would now choose the size of the training dataset for the weak learner. Recall that a weak learner should have an error ¡ 1/2 γ, with γ ¿ 0. Derive a bound for γ in terms of the sample size such that the probability of the weak learner making an error of 1/2 γ is ¿ 1 δ, for some δ ¿ 0. State any PAC learnability assumptions you make.

Question 1(d)

Give your reasoning for how you would now choose the size of the training dataset for the weak learner. Recall that a weak learner should have an error ¡ 1/2 γ, with γ ¿ 0. Derive a bound for γ in terms of the sample size such that the probability of the weak learner making an error of 1/2 γ is ¿ 1 δ, for some δ ¿ 0. State any PAC learnability assumptions you make.

Hints:

1. Recall Lecture 9 Page 3 where we talk about the implications of the Generalization Bound.
2. Lecture 7 - 9 talk about PAC learning and required assumptions specifically.

Question 1(e)

Implement your AdaBoost algorithm. Plot the accuracy (training and test) as a function of number of iterations. Plot the confidence margin, $\min_{x \in S} |h(x)|$, of your predictions vs number of iterations (5 pts). You do not need to submit your code.

Question 1(e)

Implement your AdaBoost algorithm. Plot the accuracy (training and test) as a function of number of iterations. Plot the confidence margin, $\min_{x \in S} |h(x)|$, of your predictions vs number of iterations (5 pts). You do not need to submit your code.

Hints:

1. Checkout the scikit-learn repo page for the Adaboost Classifier: `https://github.com/scikit-learn/scikit-learn/blob/c69cabda0/sklearn/ensemble/_weight_boosting.py#L341`.
2. Recall Lecture 10 Page 2 for algorithmic description of Adaboost.

Question 2(a)

Let the class conditional density $\eta(x) = P(Y = 1|X = x)$. Suppose the loss value associated with returning the reject option, 0, is c ¡ 1/2. As with the 0-1 loss, the cost of misclassifcation, with confidence, yh(x) ¡ ρ, is 1. Derive an expression for the generalization error or Bayes risk, R(h).

Question 2(a)

Let the class conditional density $\eta(x) = P(Y = 1|X = x)$. Suppose the loss value associated with returning the reject option, 0, is c ¡ 1/2. As with the 0-1 loss, the cost of misclassifcation, with confidence, yh(x) ¡ ρ, is 1. Derive an expression for the generalization error or Bayes risk, R(h).

Hints:

1. Check the papers listed in the question.
2. Do not evaluate the expectation with respect to D i.e., $R(h) = \mathbb{E}_{S\sim D}\hat{R}_S(h)$.

Question 2(c)

Bartlett and Wegkamp 2008 define the following loss:

$$h^*(x) = \begin{cases} 1 - \frac{(1-c)yh(x)}{c} & yh(x) < 0 \\ 1 - yh(x) & 0 < yh(x) < 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Note that the above loss in (2) is greater than the discontinuous loss in part (a). When $d < 1/2 \leq \rho \leq 1 - d$, they show that the excess risk with this loss for any h upper bounds the excess risk with the loss in part (a). Write down an optimization problem for the ERM of this loss (2) using bounded, affine functions, i.e., $h_{w,b}(x) = w^T x + b, |w| \leq r$. Show that this optimization is convex.

Question 2(c)

Bartlett and Wegkamp 2008 define the following loss:

$$h^*(x) = \begin{cases} 1 - \frac{(1-c)yh(x)}{c} & yh(x) < 0 \\ 1 - yh(x) & 0 < yh(x) < 1 \\ 0 & \text{otherwise} \end{cases}$$

Note that the above loss in (2) is greater than the discontinuous loss in part (a). When $d < 1/2 \leq \rho \leq 1 - d$, they show that the excess risk with this loss for any h upper bounds the excess risk with the loss in part (a). Write down an optimization problem for the ERM of this loss (2) using bounded, affine functions, i.e., $h_{w,b}(x) = w^T x + b, |w| \leq r$. Show that this optimization is convex.

Hints:

1. Recall a function f is convex if $f(tx + (1-t)x) \leq tf(x) + (1-t)f(x)$.
2. The composition of two convex functions is a convex function.

Question 2(d)

Derive the KKT conditions for the problem in part (c).

Question 2(d)

Derive the KKT conditions for the problem in part (c).

Hints: Recall

1. Slater's condition.
2. $\nabla_w L(w^*, \alpha^*) = 0$.
3. $\nabla_\alpha L(w^*, \alpha^*) = 0$.
4. Complementarity.

Question 2(e)

Implement the ERM algorithm for the problem in part (c). For this, you could start with modifying the loss function and the returned model in svm.py from class. Another option is to use a standard quadratic convex program solver. Here is an example code generated by ChatGPT – modify it to plug in the objective function and constraints derived above.

Question 2(e)

Implement the ERM algorithm for the problem in part (c). For this, you could start with modifying the loss function and the returned model in svm.py from class. Another option is to use a standard quadratic convex program solver. Here is an example code generated by ChatGPT – modify it to plug in the objective function and constraints derived above.

Hints: Use svm.py!!