CS480/680, Winter 2024

Assignment 1

Designer: Shufan Zhang; Instructor: Hongyang Zhang

Released: Jan 11; Due: Feb 8, noon

Instructions

- We do not accept hand-written submissions.
- This assignment is due by noon on Feb 8, 2024.
- For questions labelled as "**coding**", please follow the instructions provided and implement the required features. The skeleton code is provided. Unless otherwise specified, all implementations should be in *Python* using a *Jupyter Notebook*. Before submission, please make sure that your code can run without any errors. Also, be sure to save the output of each cell, as any missing output may not be graded.
- Please submit the following TWO files to LEARN:
  - A write-up in PDF format: the written answers to ALL questions, including the reported results and plots of coding questions, in a single PDF file.
  - An IPYNB file: your implementations for ALL coding questions. Please save the output of each cell, or your coding questions may NOT be graded.

Part 1: Writing (50 points)

1. **Perceptron [20 points]**

   (a) [5 points] Consider the following dataset where $\mathbf{x} = (x_1, x_2) \in \{0,1\} \times \{0,1\}, y \in \{-,+\}$. Note that this dataset is corresponding to the boolean function "AND" over the 2-bit binary input. Suppose we are training a Perceptron to learn on this dataset and we initialize $\mathbf{w}_0 = \mathbf{0}$ and $b_0 = 0$. Please i) answer if this dataset is learnable by Perceptron, and ii) if so, write down the weights update procedure for each iteration; if not, explain why.

x_1	x_2	y
0	0	-
0	1	-
1	0	-
1	1	+

   Table 1: The Truth Table for AND Function.

   (b) [5 points] Extending AND to any boolean functions over a 2-bit binary input, where we have $2^{2^2} = 16$ possible distinct boolean functions in total, among which how many of them can be learnable by a Perceptron? Please also write down the truth table(s) of the boolean functions that are **not learnable**, if there are any.

   (c) [10 points] (Modified Perceptron Algorithm) Recall the Perceptron algorithm we learned in class: for each "mistake" we update the weights by setting $\mathbf{w} \leftarrow \mathbf{w} + y_{I_t}\mathbf{x}_{I_t}, b \leftarrow b + y_{I_t}$. Now we would like to modify this algorithm by considering a constant $c > 1$ and making modifications to the update rule $\mathbf{w} \leftarrow \mathbf{w} + cy_{I_t}\mathbf{x}_{I_t}, b \leftarrow b + cy_{I_t}$. Let's call this algorithm "modified Perceptron". Please **prove or refute** that the modified Perceptron algorithm converges after the same number of mistakes as made by the Perceptron algorithm. (Hint: c.f. the convergence theorem in lecture)

2. **Regression [20 points]**

   (a) [10 points] (Variants of Logistic Regression) In class, we have shown the multi-class extension of logistic regression. The logit transformation has the form

   $$Pr[Y = k|\mathbf{x}, \mathbf{w}] = \frac{\exp(\langle \mathbf{x}, \mathbf{w}_k \rangle)}{\sum_{l=1}^{c} \exp(\langle \mathbf{x}, \mathbf{w}_l \rangle)},$$

   which is known as the Softmax function. Please derive the maximum likelihood estimation of this multi-class logistic regression.

   (b) [10 points] (Ridge Regression) The ridge regression problem is known as the linear regression that penalizes the ℓ_2 norm of the weights. That is, to solve

   $$\min_{\mathbf{W}} \frac{1}{n} \|\mathbf{W}X - Y\|_F^2 + \lambda \|\mathbf{W}\|_F^2.$$

   Please show the solution to this ridge regression problem is $\mathbf{W} = YX^T(XX^T + n\lambda I)^{-1}$.

3. **SVM [10 points]**

   (Understanding Soft-Margin SVM) As learned in class, the soft-margin SVM introduces a slack term that penalizes the misclassification of data points when the data is not linearly separable:

   $$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2 + C \cdot \sum_{i=1}^{n} s_i \quad \text{s.t. } \forall i, s_i = 1 - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b)$$

   where $s_i \geq 0$ gives the slackness for each data point in the dataset. Please answer (intuitively, no formal proof needed):

(a) [6 points] In cases that i) $s_i = 0$, ii) $s_i > 1$, and iii) $s_i \in (0, 1]$, is this data point misclassified, respectively? And where does this data point lie with respect to the decision boundary, respectively?

(b) [4 points] Please analyze intuitively the performance of the model when the parameter C increases and its relationship to hard-margin SVM.

Part 2: Coding (50 points)

Implement your own Perceptron and test with the classification task on the Iris dataset in *Python* using *Jupyter Notebook*. You are **not allowed** to use off-the-shelf libraries that already implement Perceptron algorithm. But you may use the provided skeleton code, which downloads and pre-processes the dataset and write your code based on that.
Note that the target variable to be predicted is the last feature, `class`, which describes the class of the iris flower, containing 3 possible values "Iris Setosa", "Iris Versicolour", or "Iris Virginica". So this is a multi-class classification task – you have to implement the multi-class extension of the Perceptron algorithm.

The breakdown requirements and the corresponding points for this question are shown below.
(a) [5 points] Training-test dataset splitting (80% of the data are used for training and 20% of the data for testing).
(b) [20 points] Implementation of the Perceptron algorithm.
(c) [15 points] Multi-class extension for Perceptron algorithm (either implement one vs. all or one vs. one).
(d) [10 points] Plot the accuracy ($\frac{\text{\# of correct classification}}{\text{\# of data points}}$) in each training iteration, and report the final accuracy on the test dataset.

Please train your Perceptron on the training data for **10** iterations and each iteration uses the entire training set. For more details about Perceptron and the multi-class extension, please refer to the slides used in class and the reading materials.