MACHINE LEARNING – HANOI 2019

Homework #2

Issued: May 05, 2019
Due: May 12, 2019

Problem 1. [*Logistic function*]
Let the logistic (sigmoid) function be

$$g(z) = \frac{1}{1 + e^{-z}}.$$

(a) Show that $g(-z) = 1 - g(z), \forall z \in \mathbb{R}$.

(b) Show that the inverse function of g is given by $g^{-1}(y) = \ln \frac{y}{1-y}, \forall y \in (0, 1)$.

(c) Show that $g'(z) = g(z)(1 - g(z)), \forall z \in \mathbb{R}$.

Problem 2. [*Loss function of logistic regression*]
Consider training a logistic regression on the dataset $\{\boldsymbol{x}^{(i)}, y^{(i)}\}_{i=1}^m$ using the cross-entropy loss function

$$J(\boldsymbol{\theta}) = -\sum_{i=1}^{m} y^{(i)} \log h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) + \left(1 - y^{(i)}\right) \log\left(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})\right),$$

where the hypothesis is given by

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}.$$

(a) Show that $J(\boldsymbol{\theta})$ is a convex function.

[*Hint: One way to show that a 1-D function is convex is to prove that its second derivatives is nonnegative. You may also want to use the fact that the composition of a convex function and a linear function is itself a convex function.*]

(b) Show that the gradient of J is given by

$$\frac{\partial J}{\partial \theta_j} = \sum_{i=1}^{m} (h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) - y^{(i)}) x_j^{(i)}, \quad \forall j = 0, 1, \ldots, n.$$

Problem 3. [*Duality of SVM*]
Recall the definition of the Lagrangian of a (hard-margin) Support Vector Machine:

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) := \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left[y^{(i)} \left(\boldsymbol{w}^T \boldsymbol{x}^{(i)} + b \right) - 1 \right].$$

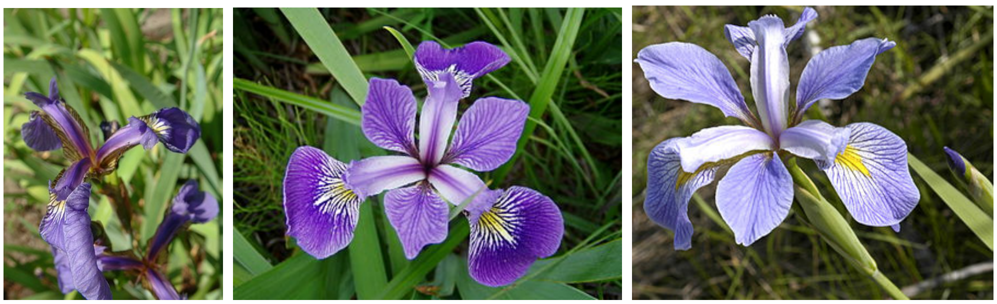
The dual problem is then defined as

$$d^* = \max_{\alpha_i \geq 0} \min_{\boldsymbol{w}, b} L(\boldsymbol{w}, b, \boldsymbol{\alpha}).$$

Show that d^* can be obtained by solving the following problem:

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle \boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)} \rangle \\
\text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, \ldots, m \\
& \sum_{i=1}^{m} \alpha_i y^{(i)} = 0.
\end{aligned}$$

Problem 4. [*Implementation of Softmax Regression and SVM*]
In this problem we use the Iris flower dataset[1] to classify 3 types of Iris: (0)-Setosa , (1)-Versicolour , and (2)-Virginica. See Fig. 1 for a visualization. The dataset contains 150 samples, each of which has 4 features: the length and the width of the sepals and petals. The ground-truth labels for all samples are also provided in the dataset. Submit a `.ipynb` file to address the following questions.



(a) Setosa (b) Versicolor (c) Virginica

Figure 1: Examples of 3 types of Iris flowers. Source: Wikipedia.

(a) Load the dataset using `sklearn` and *randomly* split it into 75 samples for training and 75 samples for testing with `random_state=0`.

(b) Train a softmax regression using `sklearn` and report the accuracy and confusion matrix on the test set.

(c) Repeat Part (b) with SVM and `linear` kernel (parameters of your choice).

(d) Repeat Part (b) with SVM and a `poly` kernel (parameters of your choice).

(e) Repeat Part (b) with SVM and `rbf` kernel (parameters of your choice).

[1] Documentation: https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html