# Homework Assignment 1
# Perceptron and Logistic Regression

Lecturer: Kyunghyun Cho

February 13, 2017

**Submission Instruction** You must typeset the answers to the theory questions using LaTeX or Microsoft Word and compile them into a single PDF file. For the programming assignment, complete the two Jupyter notebooks. Create a ZIP file containing both the PDF file and the two completed Jupyter notebooks, and name it "⟨Your-NetID⟩_hw1.zip". Email this file to `intro2ml@gmail.com` within two weeks since the announcement of the homework.

**1.** When defining a perceptron, we have augmented an input vector $\mathbf{x}$ with an extra 1:

$$M(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \tilde{\mathbf{x}}),$$

where $\tilde{\mathbf{x}} = [\mathbf{x}; 1]$. Why is this necessary? Provide an example in which this extra 1 is necessary.

SOLUTION: The model equation created after the learning process is supposed to provide a line that classifies inputs as accurately as possible. The number of variables is equivalent to $d$, or the dimension of $\mathbf{x}$, and each coefficient corresponds to the values in $\mathbf{w}$. With such an equation, there is nowhere to place a constant to account for a vertical shift in the model: it only permits models which pass through the origin. In some cases, the optimal solution after learning may be a model that does not pass through the origin, but somewhere else along the y axis. By appending a 1 to the end of $\mathbf{x}$, we can obtain an additional value from $\mathbf{w}$ during learning, which ends up being the bias. This "bias" helps shift the model by a constant and can thus provide a more optimal solution.

**2.** We used the following distance function for perceptron in the lecture:

$$D(M^*(\mathbf{x}), M, \mathbf{x}) = -\left(M^*(\mathbf{x}) - M(\mathbf{x})\right)\left(\mathbf{w}^\top \tilde{\mathbf{x}}\right).$$

This distance function has a problem of a trivial solution. What is the trivial solution? Propose a solution to this.

SOLUTION: The trivial solution is when we do not want to assign different weights to the input vector. As an example, if W is 0, then that means the distance function will always output 0, even if it shouldn't. To solve this, we need to somehow augment the weight vector to account for this situation and add a sort of intercept.

**3.** The distance function of logistic regression was defined as

$$D(y^*, \mathbf{w}, \mathbf{x}) = -(y^* \log M(\mathbf{x}) + (1 - y^*) \log(1 - M(\mathbf{x}))).$$

Derive its gradient with respect to the weight vector $\mathbf{w}$ step-by-step.

SOLUTION: Let $a = -y^* \log M(\mathbf{x})$ and $b = \log(1 - M(\mathbf{x}))$
We also know that $M(\mathbf{x}) = \sigma(\mathbf{w}^\top \tilde{\mathbf{x}})$
Based off this, we can say $a = -y^* \log(\frac{1}{1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}})$ and $b = \log(1 - \frac{1}{1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}})$
$b = \log(1 - \frac{1}{1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}}) = \log(\frac{1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}-1}{1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}}) = \log(\frac{e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}}{1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}}) = \log(e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}) - \log(1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}})$
$\frac{db}{d\mathbf{w}^\top} = \frac{-\tilde{\mathbf{x}}e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}}{e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}} + \frac{\tilde{\mathbf{x}}e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}}{1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}} = \frac{-\tilde{\mathbf{x}}(1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}})+\tilde{\mathbf{x}}e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}}{1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}} = \frac{\tilde{\mathbf{x}}}{1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}}$
$\frac{da}{d\mathbf{w}^\top} = \frac{-y^*\tilde{\mathbf{x}}e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}}{1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}}$
Bringing this together:
$\nabla_{\mathbf{w}} D(M^*(\mathbf{x}), M, \mathbf{x}) = \frac{-y^*\tilde{\mathbf{x}}e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}}{1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}} - \frac{y^*\tilde{\mathbf{x}}}{1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}} + \frac{\tilde{\mathbf{x}}}{1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}}$
$= -y^*\tilde{\mathbf{x}}e^{-\mathbf{w}^\top \tilde{\mathbf{x}}}\sigma(\mathbf{w}^\top \tilde{\mathbf{x}}) - y^*\tilde{\mathbf{x}}\sigma(\mathbf{w}^\top \tilde{\mathbf{x}}) + \tilde{\mathbf{x}}\sigma(\mathbf{w}^\top \tilde{\mathbf{x}}) = \tilde{\mathbf{x}}\sigma(\mathbf{w}^\top \tilde{\mathbf{x}})(-y^*(1+e^{-\mathbf{w}^\top \tilde{\mathbf{x}}})+1)$
$= \tilde{\mathbf{x}}\sigma(\mathbf{w}^\top \tilde{\mathbf{x}})(\frac{-y^*}{\sigma(\mathbf{w}^\top \tilde{\mathbf{x}})}+1) = \tilde{\mathbf{x}}(-y^* + \sigma(\mathbf{w}^\top \tilde{\mathbf{x}}))$
$= -(M^*(\mathbf{x}) - M(\mathbf{x}))\tilde{\mathbf{x}}$

**4.** (Programming Assignment) Complete the implementation of perceptron and logistic regression using Python and scikit-learn. The completed notebooks must be submitted together with the answers to the questions above.

Perceptron `https://github.com/nyu-dl/Intro_to_ML_Lecture_Note/blob/master/notebook/Perceptron1.ipynb`

Logistic Regression `https://github.com/nyu-dl/Intro_to_ML_Lecture_Note/blob/master/notebook/Logistic%20Regression%201.ipynb`