CS/ECE/ME532 Assignment 7

1. **Wikipedia Wisconsin PageRank**. A dataset was created by looking at the links between Wikipedia pages with the word 'Wisconsin' in the title. The corresponding graph contains 5482 nodes (Wikipedia pages) and 41,306 edges (links between the pages).

   Two files contain the data. The edges are store in a file with two columns of integers, where the first column indicates the *from node* and the second column indicates the *to node*. A second file contains the titles of the Wikipedia pages, and their integer value.

   Use the starter script to load the edges file and create an adjacency matrix $\boldsymbol{A}$, where $\boldsymbol{A}_{i,j} = 1$ if there is a edge from node j to node i, and zero elsewhere.

   a) Write code to:
   - i. ensure there are no traps by adding 0.001 to each entry of $\boldsymbol{A}$
   - ii. normalize $\boldsymbol{A}$, and
   - iii. use an eigen decomposition to rank the importance of the Wikipedia pages.

   b) What is the title of the page ranked 1st (i.e, the most important page)?

   c) What is the title of the page ranked 3rd?

2. **Gradient Descent and Logistic Regression.**

   For a binary linear classifier, the predicted class is given as $\widehat{y}_i = \text{sign}(\boldsymbol{x}_i^T \boldsymbol{w})$. Training in machine learning involves finding a $\boldsymbol{w}$ that does a good job on labeled data, and often involves solving an optimization of the form:

   $$\min_{\boldsymbol{w}} \sum_i \ell_i(\boldsymbol{w}) + \lambda r(\boldsymbol{w})$$

   where $\ell_i(\boldsymbol{w})$ is the loss on the ith training example, and $r(\boldsymbol{w})$ is a regularizer. In ridge regression, the loss function is the squared error and the regularizer is the 2-norm of $\boldsymbol{w}$, and training amounts to solving the following minimization:

   $$\min_{\boldsymbol{w}} \sum_{i=1}^{n} (\boldsymbol{x}_i^T \boldsymbol{w} - \boldsymbol{y}_i)^2 + \lambda ||\boldsymbol{w}||_2^2.$$

   For some classification tasks, the squared error can be a poor loss function, since easy-to-classify data points can have a large associated loss. This happens when a data point is correctly classified, but far from the decision boundary (i.e, the absolute value of $\boldsymbol{x}_i^T \boldsymbol{w}$ is large).

   In practice, an often more appropriate loss function is the logistic loss, given by

   $$\ell_i(\boldsymbol{w}) = \log\left(1 + e^{-y_i \boldsymbol{x}_i^T \boldsymbol{w}}\right)$$

a) For a binary linear classifier, explain (mathematically) why the logistic loss function does not suffer from the same problem as the squared error loss on easy to classify points.

b) Compute an expression for the gradient (with respect to $\boldsymbol{w}$) of the ℓ_2 regularized logistic loss:

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} \log\left(1 + e^{-y_i \boldsymbol{x}_i^T \boldsymbol{w}}\right) + \lambda ||\boldsymbol{w}||_2^2$$

c) Use the expression for the gradient that you derived to implement gradient descent and train a classifier on the provided dataset. For simplicity, you may assume $\lambda = 1$.

d) Plot the data points (indicating their class with different colors) and plot the decision boundary. What is the error rate of your classifier on the training data?

e) Train a classifier using the squared error loss, and plot the decision boundary. How does this compare with a decision boundary when trained with logistic loss?

f) Add 1000 easy to classify data points to the training set: more specifically, 1000 points with $y_i = -1$ and $\boldsymbol{x} = [10, 0]^T$. Re-train your classifiers and comment on the performance when trained with the logistic loss and the squared error.