HINT FILE: Assignment 3 (Textual Features, Logistic Regression, and Regularisation)

Artificial Intelligence (CSE-241N)
IIT (BHU) Varanasi

February 11, 2018

1 Logistic Regression

In this document we'll describe the equations for implementing a Logistic Regression model.

The predict function for logistic regression looks like:

$$y_{[n\times 1]} = sigmoid(x_{[n\times f]}W_{[f\times 1]}) \tag{1}$$

$$sigmoid(x) = \frac{1}{1-e^{-x}} \tag{2}$$

where y is the predicted values for the input data x and W is the weight vector. The dimensions of the elements are in terms of n: the number of data points and f: length of feature for each data point. We'll use $\hat{y}_{[n\times 1]}$ to represent the actual values corresponding to the data points x.

The *sigmoid* function is applied element-wise.

Now we'll give the loss equation:

$$L = \frac{1}{n}\sum_{i=1}^{n} -\hat{y}_i log(y_i) - (1-\hat{y}_i) log(1-y_i) \tag{3}$$

The gradient descent update equation is given by:

$$W_{i+1} = W_i - \eta \frac{\partial L}{\partial W} \tag{4}$$

where η is the learning rate, and

$$\frac{\partial L}{\partial W} = \frac{1}{n} x^\top (y - \hat{y}) \tag{5}$$

therefore,

$$W_{i+1} = W_i - \eta \frac{1}{n} x^\top (y - \hat{y}) \tag{6}$$

2 Regularisation

Following is the formulation for L2 regularisation for any generic loss function L,

$$\mathcal{L}(W, x) = L(W, x) + \frac{\lambda}{2f} \sum_{i=1}^{f} W_i^2 \tag{7}$$

Where λ is the regularisation parameter. The gradients of weights will change in the following way

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial L}{\partial W} + \frac{\lambda}{f} W \tag{8}$$

Therefore, the final update becomes,

$$W_{i+1} = W_i - \eta(\frac{1}{n} x^\top (y - \hat{y}) + \frac{\lambda}{f} W_i) \tag{9}$$

3 Textual Feature Extraction

3.1 Bag of Words Model

The formulation of bag of words model is given below. Let $w_1, w_2, \ldots, w_n$ be a list of unique words, then for a document d_j in a set documents $D = \{d_1, d_2, \ldots, d_k\}$, the feature vector x^j is a vector of size n is given by

$$x_i^j = \frac{frequency(w_i, d_j)}{length(d_j)} \tag{10}$$

where $frequency(w_i, d_j)$ denotes the number of times the word w_i occurs in document d_j, and $length(d_j)$ is the total number of words in d_j.

3.2 Tf–Idf Model

The formulation of Tf–Idf model is given below. Let $w_1, w_2, \ldots, w_n$ be a list of unique words, then for each document d_j in a set of documents $D = \{d_1, d_2, \ldots, d_k\}$, we calculate two vectors of size n.

The first is tf^j vector which denotes term frequency for d_j.

$$tf_i^j = \frac{frequency(w_i, d_j)}{length(d_j)} \tag{11}$$

where $frequency(w_i, d_j)$ denotes the number of times the word w_i occurs in document d_j, and $length(d_j)$ is the total number of words in d_j. The second is the idf^j vector which denotes inverse document frequency for d_j.

$$idf_i^j = log(\frac{k}{|\{d \in D : w_i \in d\}|}) \tag{12}$$

where k is the total number of documents and $|\{d \in D : w_i \in d\}|$ denotes the number of documents in which the word w_i occurs at least once.

The idf^j vector is independent of d_j, so it can be calculated in advance in a pre-processing step. Henceforth, we will denote idf^j by simply idf.

The final feature vector for each document d_j is given by

$$x_i^j = tf_i^j \cdot idf_i \tag{13}$$

Or in a vectorised notation as simply

$$x = tf \cdot idf \tag{14}$$

Note while creating the list of words $w_1, w_2, \ldots, w_n$, we discard words which occur for a total of less than μ times, μ is a parameter of the model. The rationale behind this is that rare words increase the sparsity of the feature vector and provide no new information as they are mostly rare named entities.