**Bayesian Generalised Linear Model**

**Overview:**

In this project we will be working with Generalized Linear Models (specifically, the simpler version with canonical link function), including Logistic regression, Poisson regression and Ordinal regression. Our goal is to use one generic implementation for the main algorithm that works for multiple observation likelihoods.

**Data:**

Each dataset is given in two files with the data in one and the labels in the other file. We will use the datasets A and usps for classification with logistic regression. We will use the dataset AP for count prediction with Poisson regression. We will use the dataset AO for ordinal prediction with ordinal regression.

The datasets A, AP, AO were artificially generated with labels which are not perfectly matched to any linear predictor yet they are generated to be somewhat predictable. The examples in usps represent 16×16 bitmaps of the characters 3 and 5 and are taken from the well known usps dataset (representing data originally used for zip code classification).

**Implementing our variant of GLM**

In this project, we will use a Bayesian (or regularized) version of the algorithm with w ∼ N (0, 1/α \* I), where α = 10, to calculate the MAP solution wMAP .

* Logistic regression (and by extension GLM) relies on a free parameter (w0) to capture an appropriate separating hyperplane. Therefore, we will need to add a feature fixed at one (also known as an intercept) to all datasets in the assignment. To match the test case below we add this as the first column in the data matrix.
* The vector of first derivatives of the log posterior is:  
  where d is a vector whose elements are di .

A screenshot of a computer

Description automatically generated with low confidence

**Likelihood models:**

Text, letter

Description automatically generated

* For the logistic model the update formula is:

wn+1 ← wn − (−αI − Φ T RΦ)−1 [ΦT (t − y) − αwn]

A picture containing text, person, document

Description automatically generated

**Implementation and Evaluation of the Algorithms**

The main idea in GLM is to build a generic implementation that in principle can handle any likelihood model. Accordingly we create one implementation of the optimization and evaluation process which makes use of functions that compute the 4 items described in the last section. We then call this implementation on each likelihood model to obtain results.

Our goal is to implement the GLM algorithm and generate learning curves with error bars (i.e., ±1σ) as follows.

Repeat 30 times :

Step 1) Set aside 1/3 of the total data (randomly selected) to use as a test set.

Step 2) Permute the remaining data and record the test set error rate as a function of increasing training set portion (0.1,0.2, . . . ,1 of the total size).

We then calculate the mean and standard deviation for each size and plot the result. In addition, we also record the number of iterations and the run time until convergence in each run and then we report their averages.

**Info about the code:**

1. The code is divided into three sections: Part 1: Logistic Regression, Part 2: Poisson Regression and Part 3: Ordinal Regression.
2. In each part, the code divides the datasets into Train and Test Dataset and pass them as arguments along with other required values to the common functions created to perform the predictions.
3. Out of these 3 sections of code, one is left uncommented and the remaining three are commented.

**Steps to run the code:**

1. Change the path values in the path variables according to the new location where the code will be run. The variables are divided into the dataset and the respective label file names of the datasets.

**Text

Description automatically generated**

1. Move to the part which you want to execute and uncomment the code for that particular model that you want to test the code on.

**Text

Description automatically generated**

1. For Logistic Model:

This model runs on two datasets: A.csv and USPS.csv

Initially, A.csv is being used by default. In order to use the USPS dataset, just replace the dataset path variable accordingly to USPS path variable**.**

Text

Description automatically generated

1. Run the code.
2. The plots will be automatically shown at the end of the execution.