# Lab 5 – Logistic Regression

MACHINE LEARNING

SAUMAY AGRAWAL
16BCE1151

## EXPERIMENT

- Implement the algorithm, and run it for the seeds  dataset and then for your chosen dataset.

## ALGORITHMS

### LOGISTIC REGRESSION

It's a classification algorithm, that is used where the response variable is categorical. The idea of Logistic Regression is to find a relationship between features and probability of particular outcome. This type of a problem is referred to as Binomial Logistic Regression, where the response variable has two values 0 and 1 or pass and fail or true and false. Multinomial Logistic Regression deals with situations where the response variable can have three or more possible values. We know that, Linear regression model can generate the predicted probability as any number ranging from negative to positive infinity, whereas probability of an outcome can only lie between 0 and 1. Also, Linear regression has a considerable effect on outliers. To avoid this problem, logit function is used. Sigmoid function is the inverse of logit function, which restricts the outcome values to be between 0 and 1. Thus we are able to classify, whether a set of input features belong to a particular class or not.

### SOLVERS OF LOGISTIC REGRESSION

**Newton's Method (newton-cg)**
Newton's method uses quadratic function (cost function) minimisation which is better than the general gradient descent algorithm because it uses the quadratic approximation (i.e. first and second partial derivatives). Moreover, the geometric interpretation of Newton's method is that at each iteration one approximates f(x) by a quadratic function around xn, and then takes a step towards the maximum/minimum of that quadratic function (in higher dimensions, this may also be a saddle point). Note that if f(x) happens to be a quadratic function, then the exact extremum is found in one step.

Drawbacks:

- It's computationally expensive because of The Hessian Matrix (i.e. second partial derivatives calculations).
- It attracts to Saddle Points which are common in multivariable optimization (i.e. a point its partial derivatives disagree over whether this input should be a maximum or a minimum point!).

**Limited-memory Broyden–Fletcher–Goldfarb–Shanno Algorithm (lbfgs)**
In a nutshell, it is analogue of the Newton's Method but here the Hessian matrix is approximated using updates specified by gradient evaluations (or approximate gradient evaluations). In other words, using an estimation to the inverse Hessian matrix. The term Limited-memory simply means it stores only a few vectors that represent the approximation implicitly. When the dataset is small, L-BFGS relatively performs the best compared to other methods especially it saves a lot of memory, however there are some "serious" drawbacks such that if it is unsafeguarded, it may not converge to anything.

**A Library for Large Linear Classification (liblinear)**
It's a linear classification that supports logistic regression and linear support vector machines (A linear classifier achieves this by making a classification decision based on the value of a linear combination of the characteristics i.e feature value). The solver uses a coordinate descent (CD) algorithm that solves optimization problems by successively performing approximate minimization along coordinate directions or coordinate hyperplanes. It applies Automatic parameter selection (a.k.a L1 Regularization) and it's recommended when we have high dimension dataset (recommended for solving large-scale classification problems).

Drawbacks:

- It may get stuck at a non-stationary point (i.e. non-optima) if the level curves of a function are not smooth.
- Also cannot run in parallel.
- It cannot learn a true multinomial (multiclass) model; instead, the optimization problem is decomposed in a "one-vs-rest" fashion so separate binary classifiers are trained for all classes.

**Stochastic Average Gradient (sag)**
SAG method optimizes the sum of a finite number of smooth convex functions. Like stochastic gradient (SG) methods, the SAG method's iteration cost is independent of the number of terms in the sum. However, by incorporating a memory of previous gradient values the SAG method achieves a faster convergence rate than black-box SG methods.
It is faster than other solvers for large datasets, when both the number of samples and the number of features are large.

Drawbacks:

- It only supports L2 penalization.

- Its memory cost of O(N), which can make it impractical for large N (because it remembers the most recently computed values for approx. all gradients).

**SAGA**

The SAGA solver is a variant of SAG that also supports the non-smooth penalty=l1 option (i.e. L1 Regularization). This is therefore the solver of choice for sparse multinomial logistic regression and its also suitable very Large dataset.

# OBSERVATIONS

## SEEDS DATASET

- Upon pairplotting the attributes of the seeds dataset, I found out that V4 and V6 attributes had clearly defined boundaries for the classification.
- Keeping the solver as the default 'liblinear', I increased the value of C, which is the inverse value of regularisation strength in this function. I found out that accuracy of the model increased with the increase in value of C.
- Even as the value was C was increased and decision boundaries were plotted, the shift in decision boundaries was found to be in the direction to maximize accuracy.
- For any given value of C, the default 'liblinear' solver was found to be the worst performing model. But it also took the least fitting time among all the solvers, in almost all the cases.

## OLYMPICS DATASET

- I reduced the dataset to only the athletes which are winners in the game of badminton, and pairplotted it. I found out that the attributes 'Height' and 'Weight' were somewhat classifiable.
- Similar trend was found between the value of C and accuracy of the model with the Olympics datasets. However, for the largest dataset, ie 120 years of data, the trend deviated. For this dataset, the highest accuracy was found to be with C=1.
- For any given value of C, the 'newton-cg' solver took the most time for fitting. Meanwhile, accuracy wasn't affected by the type of solver.
- I tried to compare the results for entire dataset using GridSearchCV(), so as to verify the performances of the 'sag' and 'saga' solvers with the others, on larger datasets. However, this operation exceeded my system's resources.

# INFERENCE

- The logistic regression algorithm is a descent algorithm for the classification of categorical data.
- It works very well if the decision boundaries are clearly evident in the dataset. But it won’t thrive when the dataset requires very complex decision boundaries for accurate classification.
- The accuracy of the model, in general, increases with an increase in the value of C, ie with the decrease in the strength of regularization. But it may vary from dataset to dataset.
- Newton’s Method is computationally expensive, among all the solvers, because of the second partial derivatives calculations.