Week 08 - GLM Investigation

Greeshma Poli

Saint Louis University

Course: High Performance Computing

March 30th, 2025

Module/Framework/Package	**Name and Brief Description of the Algorithm**	**Example of a Situation Where This GLM Implementation is Superior to Base R or Equivalent in Python**
Base R (in the stats library)	Python's stats models method is comparable to Base R's glm() function in the stats package. The model uses the Iteratively Reweighted Least Squares (IRLS) method to determine parameters in GLMs.	The algorithm functions best with datasets of small to medium size because it faces limitations when processing large datasets. The execution of logistic regression on 50,000 observation data points remains possible yet its performance declines substantially when dealing with millions of observations.
Big data version of R	Frequently used R packages for large data is Data.table, ff, and bigmemory are Helped to maximize memory utilization.Various optimizations such as chunk-wise processing, sparse matrix optimizations, and modified IRLS.	The system handles extremely large datasets which exceed the available memory capacity. Base R programming strategies would experience failures dealing with healthcare claims analysis on datasets containing millions of records because of insufficient memory storage.
Dask ML	The system implements parallel and distributed computing to run gradient descent-based solvers such as L-BFGS as part of its solution approach.	This system demonstrates excellence in dealing with large-scale out-of-core computations which includes processing enormous e-commerce transaction logs. For example, predicting customer churn for an online retail company with millions of users in real time.
Spark R	The sparklyr package in Spark R offers an interface to the spark.glm() function in Spark MLlib, enabling effective distributed GLM	This tool suits applications that need to process extremely large datasets which require more memory than what a single machine

	processing. Fisher Scoring constitutes a second-order optimization method that outperforms IRLS during distributed processing because it executes faster in distributed setups. The system processes big data by using Spark to achieve efficient distributed data processing.	can provide. An advertisement platform which monitors user interactions between different websites uses clickstream data analysis as an example.
Spark optimization	Supports L-BFGS (Limited-memory Broyden–Fletcher–Goldfarb–Shanno) and Gradient Descent optimization methods.The optimization implementation in Spark MLlib includes the function \`ml_logistic_regression()\` from \`sparklyr\` and \`LogisticRegression\` in the PySpark environment. Large datasets benefit from L-BFGS because it performs efficient approximations of Hessian matrices.	Large GLM training tasks that need to handle extremely large datasets occur best in Hadoop-based cloud environments. An example usage occurs when detecting fraudulent credit card transactions across billions of records distributed throughout the system.
Scikit-learn	The algorithm includes three optimization methods Coordinate Descent, L-BFGS and SGD which let users select based on their problem dimensions. The package integrates two regularization methods including L1 (Lasso) and L2 (Ridge) among its components.	The program functions more flexibly and delivers optimal results based on dataset dimensions. Effective for high-dimensional sparse data such as genomic data analysis. The optimization approaches from Scikit-learn demonstrate better performance than Base R during genetic dataset analysis because they offer improved scalability features and regularization capabilities.