Project 3: Logistic Regression with ADMM

Aim

The project's goal is to use the ADMM technique with distributed computation to apply logistic regression on a sizable, divided dataset. One of the objectives is to establish consensus estimates for coefficients (intercept and explanatory variables) across partitioned data by using logistic regression with ADMM, utilizing mpi4py to leverage parallel processing on the SeaWulf cluster, where a single MPI process manages a single data file, processing several files while avoiding the creation of any intermediary files and submitting a job submission SLURM file for execution along with reproducible scripts.

Code Logic

Basically core implementation is contained in the Python script file = l**ogistic_regression_admm**.py. The SLURM batch job configuration used to run the script in a cluster environment is stored in run_admm.slurm

Logistic_regression_admm.py:

- MPI Initialization: The script uses mpi4py to initialize MPI communication (comm), acquiring the number of processes (size) and the rank of each process (rank).
- Data Loading and Distribution:
    - The process with rank 0 collects the list of all data files from a specified directory (data_dir), e.g., /gpfs/projects/AMS598/projects2025_data/project3_data/.
    - Files are evenly distributed across MPI processes to balance workload (files_per_process).
    - Each process loads its assigned files, reading response variable y (binary 0/1) and explanatory variables x1 ... x25.
    - An intercept column is added to features (1s column).
- Feature Standardization:
    - Features except the intercept are globally standardized (mean 0, variance 1) across all processes. MPI collective communications (Allreduce) compute global statistics to ensure consistent scaling.
- ADMM Variables:
    - Local coefficient vectors (x_local_list) initialized to zero.
    - Global consensus variable z initialized to zero.
    - Dual variables u_local_list initialized to zero.

- Penalty parameter rho influences the strength of consensus enforcement.

- ADMM Iteration Loop:
  - Each process performs a local update for its local variables using Newton's method to minimize logistic loss combined with a quadratic penalty (adjustment from ADMM).
  - Updated local estimates plus dual variables are collectively averaged (Allreduce) to produce a new global consensus variable z.
  - Primal and dual residuals are computed and aggregated across processes to assess convergence.
  - Dual variables are updated based on deviations from consensus.
  - Loop continues until residuals fall below a set tolerance or maximum iterations (max_iterations=150) are reached.

Run_admm.SLURM:

This script automates job submission to the cluster environment:

- Specifies 1 node and 10 tasks (processes), with resource allocations (CPUs, memory, time) tailored to run a job using 10 MPI ranks.
- Loads necessary modules: Python 3.11.2 and the latest mpi4py.
- Prints job metadata: start time, job ID, nodes, number of tasks, and working directory.
- Runs the main MPI program:

```
mpirun -n $SLURM_NTASKS python
-u logisticregressionadmm.py
```

Execution Command

```
cd /gpfs/home/gdeshpande/assignment3 && sbatch run_admm.slurm

squeue -u gdeshpande

$ cat
/gpfs/home/gdeshpande/assignment3/admm_logistic_1455286.out
```

Results and Analysis

```
Job ID: 1455286
Running on nodes: dg014
Number of tasks: 10
Working directory: /gpfs/home/gdeshpande/assignment3
Okay! Found 10 data files
Now, Using 10 MPI processes
So, the Standardizing features...
So, the Features standardized successfully
Now we are starting ADMM iterations...
So, the number of features are: 26
Iteration 5: Primal residual = 0.005039, Dual residual = 0.186434
Iteration 10: Primal residual = 0.002759, Dual residual = 0.105622
Iteration 50: Primal residual = 0.000033, Dual residual = 0.001488
Converged after 54 iterations!
The Intercept: 0.266496
Explanatory Variables:
  x 1:   0.096642
  x 2:   0.006057
  x 6:  -0.004982
  x22:  -0.104751
  x23:   0.001679
  x24:   0.006089
  x25:   0.250197
So the total samples processed: 10000000
```

So, the job started and the cluster assigned a node dg014.

- 10 data files were found and assigned one each to 10 MPI processes.
- Features were standardized successfully across files/processes.

- ADMM iterations are logged showing progressive decrease in primal and dual residual errors, indicating iterative convergence.
- Convergence was achieved after 54 iterations, well within the maximum allowed iterations.
- Final logistic regression coefficients (including intercept) are output for all features.
- Massive dataset processed: 10 million rows split across 10 files.
- Script ran in parallel efficiently using MPI.