Project 2
SPM course a.a. 24/25

May 17, 2025

Distributed Parallel Random Forest

Design and implement a scalable, parallel, and distributed version of the Random Forest algorithm for classification. Consider "large" datasets stored in CSV format containing numerical features and binary labels. Your solution should load the dataset from files and write the results (predicted labels for each instance) into output files. Use input datasets from the UCI repository (e.g., SUSY, Covertype, Iris): https://archive.ics.uci.edu/ml/index.php

Tasks

1. Single-node versions (shared-memory)
   Implement two shared-memory parallel versions of the Random Forest algorithm, one using FastFlow and one using OpenMP pragmas. Both implementations must provide the same classification results for the same input dataset.
   You could base your sequential implementation starting from one of the online available C++-only implementations of the Random Forest algorithm, provided you explicitly report it in the source code and the PDF report. Demonstrate the correctness of your Random Forest implementation by comparing it to scikit-learn on a small dataset (e.g., Iris or a reduced SUSY subset) using the same train/test split, and report metrics such as accuracy and macro-averaged F1-score.
2. Multi-node hybrid version
   Implement a distributed version of the Random Forest algorithm, combining MPI with either FastFlow or OpenMP (clearly justify your choice). Parallelize both the training phase (construction of decision trees) and the classification phase (prediction using the trained model).
3. Performance evaluation
   Evaluate your parallel and distributed implementations by explicitly varying the following parameters: a) number of decision trees; b) dataset size and characteristics (i.e., different numbers of samples and features); c) the number of FastFlow/OpenMP threads per node.
   Provide single-node speedup and efficiency curves, as well as strong and weak scalability curves on the SPM cluster (up to 8 nodes). Evaluate also the training and prediction phases separately.
   Explicitly discuss the effect of changing the number of MPI processes and the number of threads per MPI process.
4. Analysis
   Provide an approximate cost model of your distributed Random Forest implementation. Clearly identify performance bottlenecks, describe challenges encountered during implementation, and summarize any optimizations applied.

All parallel implementations must aim at minimizing parallelization overhead and maximizing resource utilization.

Deliverables

Provide all source files, scripts to compile and execute your code on the cluster nodes, and a PDF report (max 15 pages) describing your implementations and the performance analysis conducted. Mention the challenges encountered and the solutions adopted. Submit by email to massimo.torquati@unipi.it your source code and PDF report in a single zip file named 'SPM_project2_<YourName>.zip'**.** Please use the email subject "SPM Project".