README for R Simulation and Prediction Script

Chinedu Nzekwe

2024-04-09

Overview

This script conducts a comprehensive simulation and evaluation of various statistical models for interaction selection and prediction in R. It leverages a broad array of packages for data manipulation, statistical modeling, and parallel computing. The primary aim is to compare the performance of different modeling techniques on simulated datasets, which mimic complex real-world data scenarios.

Dependencies

To run this script, you need to install the following R packages:

- `tidyverse`: For data manipulation and visualization.
- `MASS`: Provides functions and datasets to support Venables and Ripley's MASS.
- `mvnfast`: For multivariate normal and Student's t distributions.
- `Matrix`: For sparse and dense matrix classes and methods.
- `caret`: For data splitting, pre-processing, feature selection, model tuning using resampling.
- `glmnet`: For LASSO and elastic-net regularized generalized linear models.
- `ncvreg`: For fitting regularization paths for linear regression, logistic and multinomial regression models, Poisson regression, and the Cox model.
- `RAMP`: For regularized regression modeling, which is particularly useful for high-dimensional data.
- `iRF`: Implements iterated random forests for detecting stable high-order interactions. **For this study we used iRF 2.0.0 (`devtools::install_github("karlkumbier/iRF2.0")`)**, which was working only on MacOS and Linux. The new version of iRF (`iRF3.0.0`) works on both Windows and MacOS.
- `randomForest`: For classic random forest algorithms.
- `ranger`: A faster implementation of random forest.
- `Boruta`: For all-relevant feature selection.
- `mccr`: For computing the Matthews correlation coefficient.
- `parallel`: For parallel computing capabilities.

Additionally, `SimDesign` package is implied for generating multivariate normal samples, though it needs to be loaded explicitly.

Core Functionality

1. Data Simulation (DF.Gen): Generates synthetic datasets based on predefined models from literature. It creates variables and interactions to explore model performance under various conditions.

2. Prediction Metrics (mse.fun): A function for calculating the Mean Squared Error (MSE) between predicted and actual values.

3. Interaction Selection Algorithms:

   - LASSO (`LASSO.func`)
   - Non-Convex Penalty (`NCP.func`)
   - RAMP Algorithm (`RAMP.func`)
   - Random Forest (`RF.func`)
   - Iterative Random Forest (`iRF.func`)

4. Simulation Control (`sim.func`): Orchestrates the simulation process, calling data generation, model fitting, and evaluation functions. It iterates through various models and parameters to assess performance.

5. Results Aggregation and Reporting: Summarizes the outcomes from multiple simulation iterations, providing insights into model accuracy, variable selection performance, and computational efficiency.

Execution Guide

1. Ensure all required packages are installed and loaded.

2. Define simulation parameters (e.g., sample size, number of predictors, model configurations).

3. Call the main simulation function (sim.func) with the desired parameters.

4. Use the provided functions to evaluate and compare the performance of different modeling approaches.

Post-Simulation Analysis

The script includes functions for detailed analysis of model performance, including sensitivity, specificity, false positive rates, false negative rates, and L2 norm evaluations. It also includes functionality for summarizing the results in a structured manner for reporting and further analysis.

Output Analysis

- Output analysis focuses on both prediction accuracy (MSE) and the ability of models to correctly identify relevant predictors and interactions.

- Predictive accuracy and error metrics for both training and test datasets.

- Sensitivity, specificity, and other relevant classification metrics.

- Matthews correlation coefficient (MCC) as a balanced measure of binary classification performance.

Notes

- The simulation environment is set up to leverage parallel computing capabilities to enhance computational efficiency.
- The script is designed to be modular, allowing for easy extension or modification to include additional models or evaluation metrics.
- These results are compiled for each model and interaction selection method, providing a comprehensive overview of their strengths and limitations in handling binary outcome data.