Methods for Subsampling: Towards a Realistic Evaluation

Charlie (Changrui) Liu

University of Kentucky, Department of Statistics

Charlie.Liu@uky.edu

June 4, 2020

Overview

- Introduction
- Description of Algorithms
- Simulation Set-Ups, Results and Discussions
- Possible Future Works

Introduction and Motivation

- Today's society is producing exponentially more massive datasets; literally every field and industry is data-based. Historically speaking, the cumulative collection of data can only grow even larger
- While the amount of data seems to be limitless, our computing power is still limited; this is especially true for researchers and scholars who do not have access to super-computers
- What should we do when there is a lack of computational power? i.e. when the dataset is so big that our computational device cannot handle?

Introduction

Below are several approaches introduced in the literature in order to make the analysis scalable:

- Engineering solutions
- Statistical solutions
  - Subsampling (Use the subsample as a surrogate to perform analysis) - select a subsample of observations
  - Selection of a subsample of covariates
  - "Divide and Conquer" and etc. (Meng et al. 2017)
- We will be focusing on subsampling
- Our goal of this research is to do a comparison of realistic (not asymptotic) computational time and accuracy of some of the subsampling methods

Introduction of Notations

- $\boldsymbol{X}$ - $n \times p$ model matrix of covariates based on full sample data
- $\boldsymbol{y}$ - $n \times 1$ response vector based on full sample data
- β - $p \times 1$ vector containing all regression coefficients
- $\boldsymbol{\epsilon}$ - $n \times 1$ vector of random error terms
- Throughout, we assume an underlying linear regression model

$$\boldsymbol{y} = \boldsymbol{X}\beta + \boldsymbol{\epsilon}$$

Algorithms (Overview)

Below is a list of subsampling methods that we considered

- (Baseline) Full Ordinary Least Squares (OLS) and Uniform Subsampling (*UNIF*)
- Leverage-based Subsampling (Shrinkage Leverage *SLEV* and Binary-Fast Shrinkage Leverage *BFSLEV*)
- Information-Based Subsampling (Information-Based Optimal Subdata Selection *IBOSS* and Ratio-Based Informative Subsampling *RBIS*)
- Optimal-Design-Based Subsampling (omitted)

A Visualization of 4 Algorithms

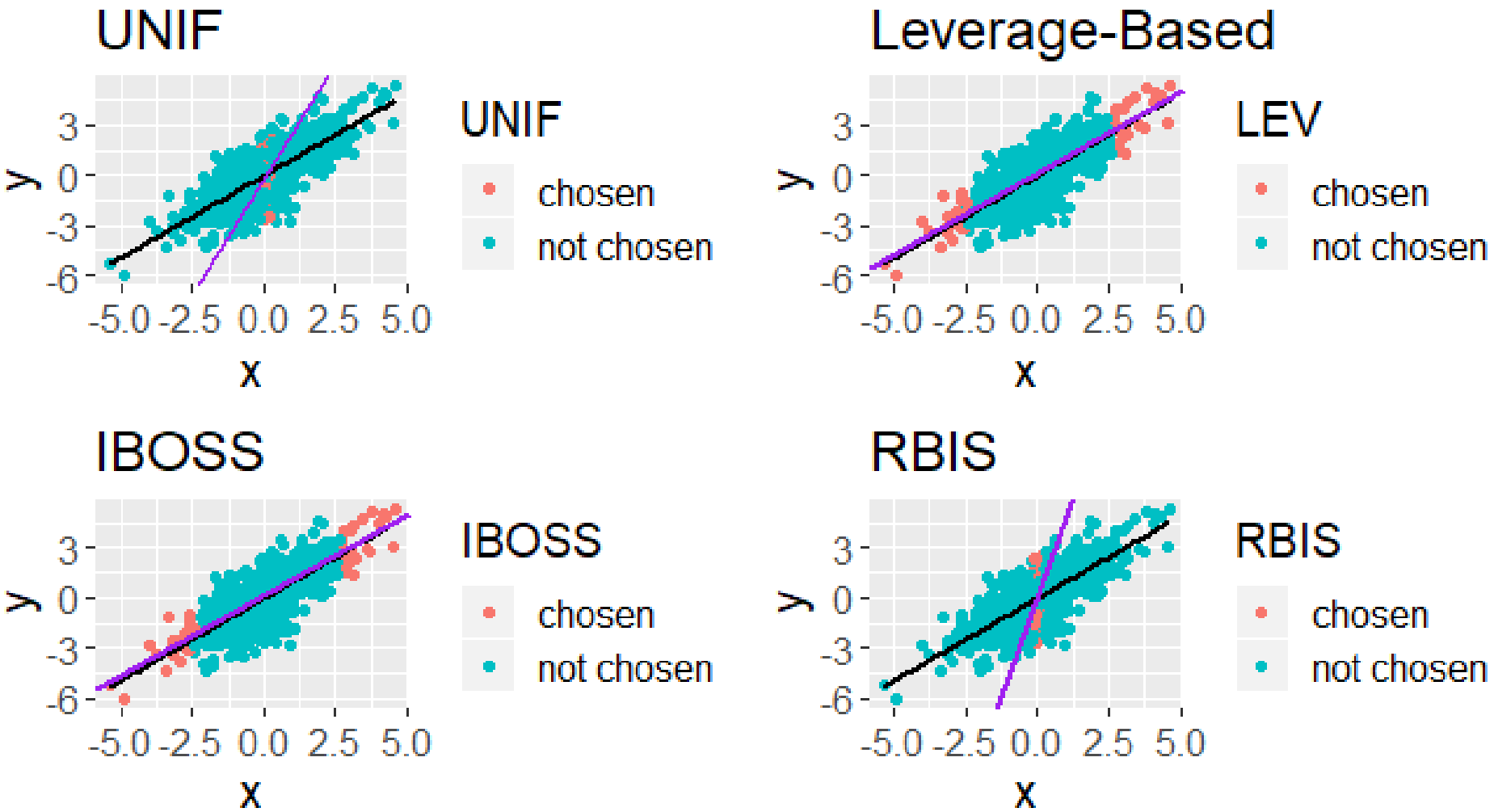

Figure: A visualization of different algorithms

Algorithms (Full OLS and UNIF)

- To calculate the full OLS, simply apply the formula

$$\hat{\beta}_{\text{OLS}} = (X^T X)^{-1} X^T \boldsymbol{Y}$$

- For uniform subsampling, simply sample r observations such that each observation is chosen with probability $\pi_i = \frac{1}{n}$
- After getting the subsample, simply perform OLS on the subsample, using it as a surrogate of the full sample
- Asymptotic computation time: $O(np^2)$ for full $O(rp^2)$ for *UNIF* (Drineas et al. 2012)
- Critique: Fairly large chance to get a pretty bad subsample when downsampling too hard and when the distribution of leverage score is non-uniform (Ma et al. 2015)

Algorithms (Leverage Scores and Leverage-Based Subsampling)

- $\hat{\boldsymbol{y}} = X\hat{\beta}_{\text{OLS}} = H\boldsymbol{y}$ where $H = X(X^TX)^{-1}X^T$ and is called the hat matrix
- The diagonal entries of H, denoted as $h_{11}, \ldots, h_{nn}$ are called leverage scores
- Leverages scores have been shown to provide important information on model fits; high leverage points tend to have high potential influencing linear model (Kutner et al. 2004)
- To perform leverage-based algorithm (LEV), use $\pi_i = h_{ii}$ as the subsampling probability distribution to draw a subsample of size r (in a sense that higher leverage points are more likely to be chosen)
- After getting the subsample, perform a Weighted Least Squares (WLS) analysis on the subsample with weights $\frac{1}{\sqrt{r\pi_i}}$ (Ma et al. 2015)

Algorithms (Shrinkage Leverage)

- It has been shown that very small leverage scores can sometimes inflate the variance to arbitrarily large (Ma et al. 2015); a fix to this is to use Shrinkage Leverage ($SLEV$) in place of purely leverage-based subsampling
- To employ $SLEV$, we use subsampling probability distribution $\pi_i = 0.9 \cdot \frac{h_{ii}}{p} + 0.1 \cdot \frac{1}{n}$; this is a mixture of $UNIF$ and LEV to get a subsample of size r (Ma et al. 2015)
- After getting the subsample, calculate the WLS on the subsample with weights $\frac{1}{\sqrt{r\pi_i}}$
- Unfortunately, the approximate computational complexity for computing the leverage score is $O(np^2)$ (given that $r << n$), which is in the same scale as full OLS
- SLEV itself is not computationally attractive

Algorithms (Approximate Shrinkage Leverage)

- Drineas et al. 2012 and Ma et al. 2015 depicted a method that could calculate the approximate version of leverage score in $o(np^2)$ time (given that $r << n$), called Binary-Fast (BF) approximation
- To employ *BFSLEV*, first calculate the approximate leverage scores, then use the approximate leverages scores in place of the exact leverage scores in the previous *SLEV* algorithm
- Please refer to details in Drineas et al. 2012 and Ma et al. 2015

Algorithms (Information-Based Optimal Subdata Selection)

- Wang et al. 2019 introduced IBOSS method, which is based on D-Optimality criterion
- To employ *IBOSS*, first sort each column in the covariate matrix, then find the observations with $\lceil \frac{r}{2(p-1)} \rceil$ smallest values and another $\lceil \frac{r}{2(p-1)} \rceil$ largest values from each column
- Take all points from the previous step and get rid of any repetitions; perform OLS on the resulting subsample
- *IBOSS* further reduced the asymptotic computational complexity to $O(np)$, given that $r << n$; most time spent is sorting

Algorithms (Ratio-Based Informative Subsampling)

- Inspired by *IBOSS*, *RBIS* follows a similar procedure, with additional usage of information from $\boldsymbol{y}$
- To employ *RBIS*, first sort each covariate column i with respect to the ratio of $\frac{\boldsymbol{y}}{\boldsymbol{X_i}}$, then find the observations with $\lceil \frac{r}{2(p-1)} \rceil$ smallest values and another $\lceil \frac{r}{2(p-1)} \rceil$ largest values from each column
- The ratio idea roughly comes from the fact that the estimation $\hat{\beta}$ somehow resembles the ratio of y and X, so taking the ratio may give us some more information about $\hat{\beta}$ without much increase in computational time to see how do we best take subsample under different circumstances
- Given that $r << n$, the asymptotic computational complexity of *RBIS* is $O(2np)$; most time spent is sorting and doing the division

To Ensure a Fair Comparison

- All of the following tests are conducted in **R**
- All of functions and codings are done solely by myself
- None of `C`-based codes (e.g. the embedded `lm` function in **R**) are used; all codings are rather naive, but fairly comparable
- Further improvement and more efficient coding can certainly be expected; please contact me if you are interested

Simulation Set-Ups

- $\boldsymbol{X}$ is generated from one of the following 3 distributions: multivariate normal distribution (very uniform leverage scores), multivariate T_5 distribution (slightly non-uniform leverage scores), and multivariate T_1 distribution (very non-uniform leverage scores) (Ma et al. 2015)
- β is a vector of preset values (mostly 1 and 0.5 and several 0's), designated to contain some sparsity
- Error terms are assumed to follow a $\mathcal{N}(0, 0.25I_n)$ distribution
- $\boldsymbol{y}$ is generated using $\boldsymbol{y} = \boldsymbol{X}\beta + \boldsymbol{\epsilon}$

Initial Simulation

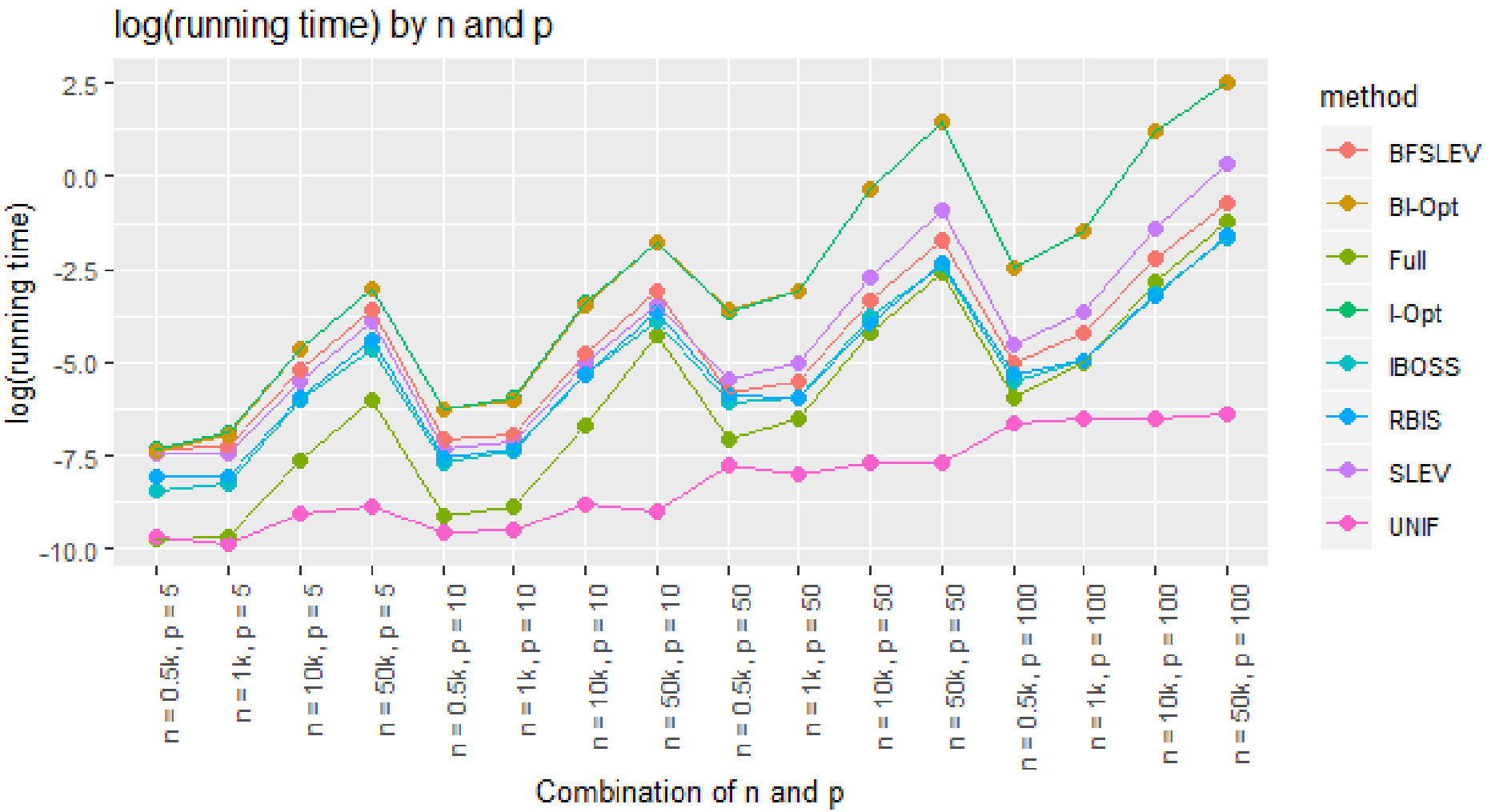

Figure: A comparison of log running times using all algorithms with fixed $r = 200$

Initial Simulation (Continued)

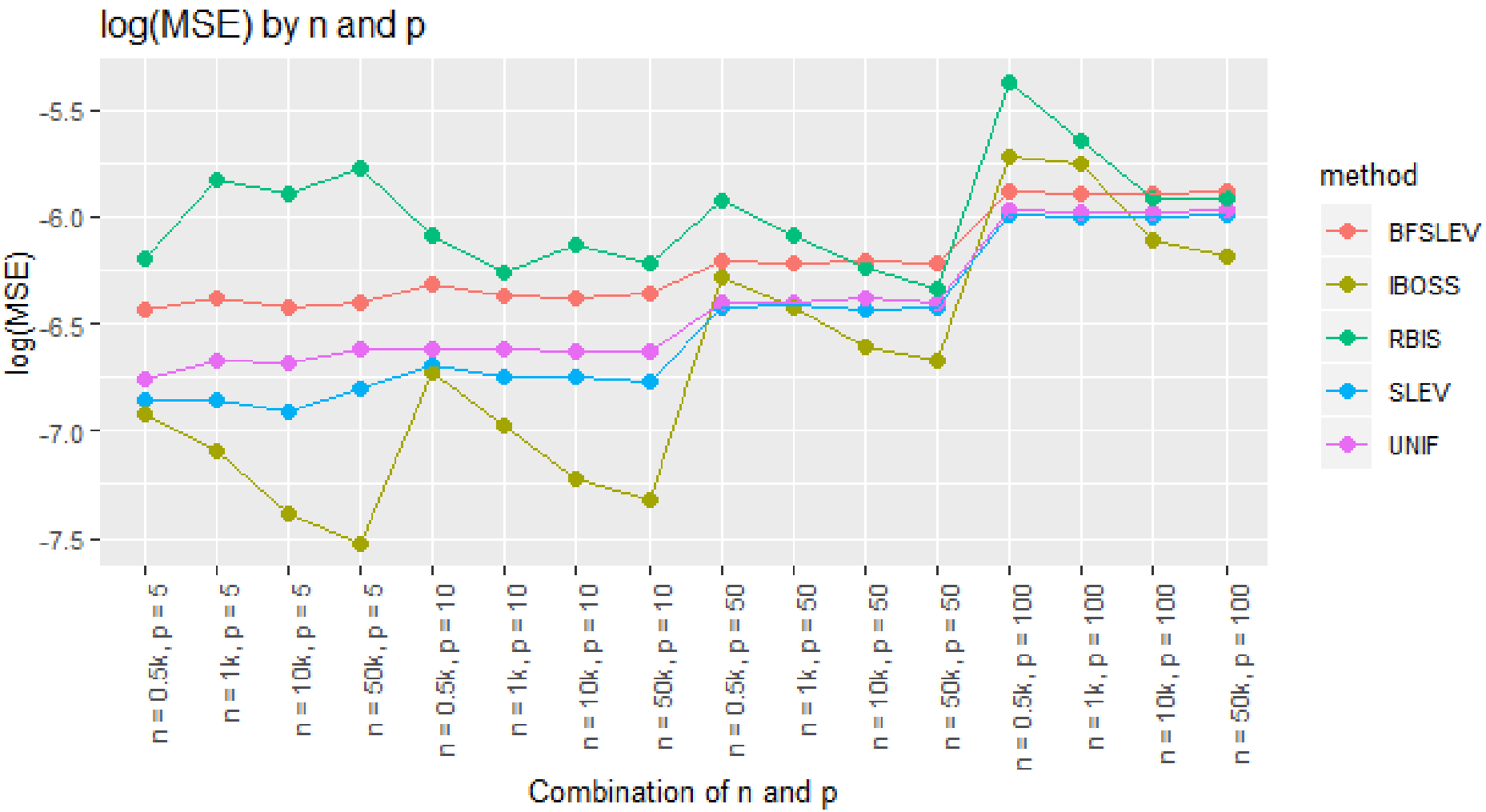

Figure: A comparison of log MSE using all algorithms with fixed $r = 200$

Simulation #1 fix $n = 50k$; vary p with $r = 1.5p$

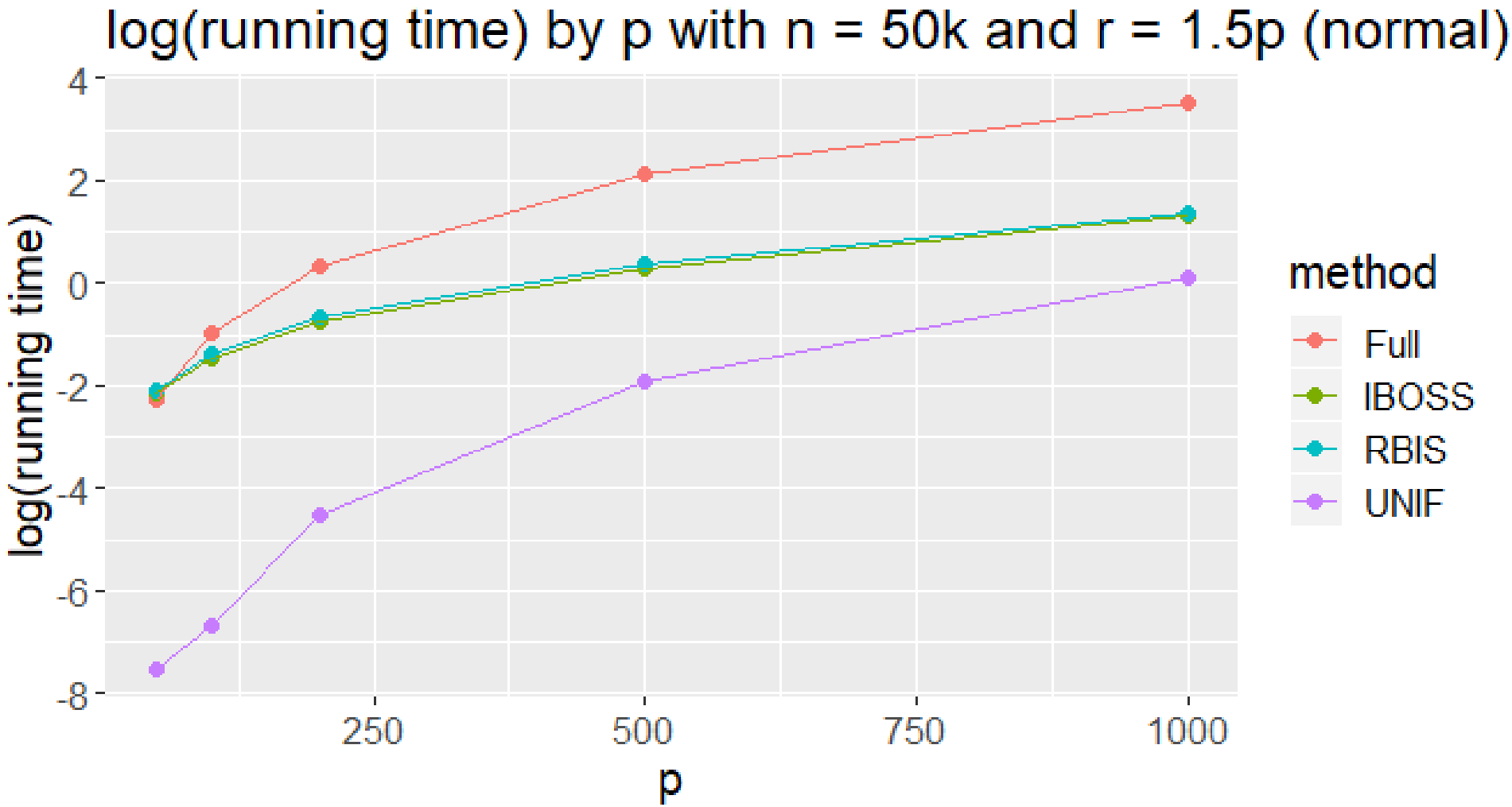

Figure: Time plot (promising algorithms only) with $n = 50k$ and $r = 1.5p$

Simulation #1 fix n; vary p with $r = 1.5p$

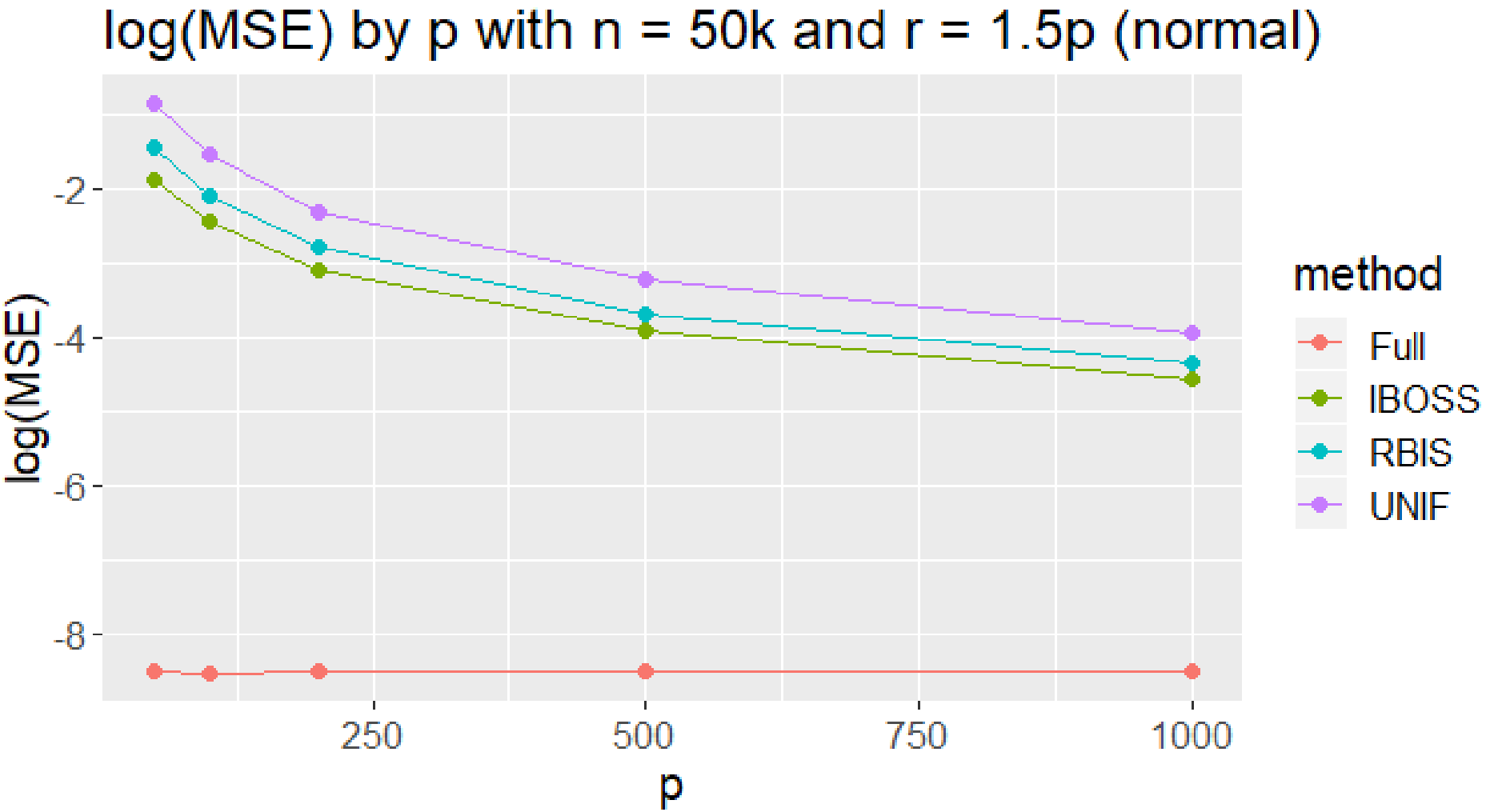

Figure: MSE plot (promising algorithms only) with $n = 50k$ and $r = 1.5p$

Simulation #2 fix n and p with varying r

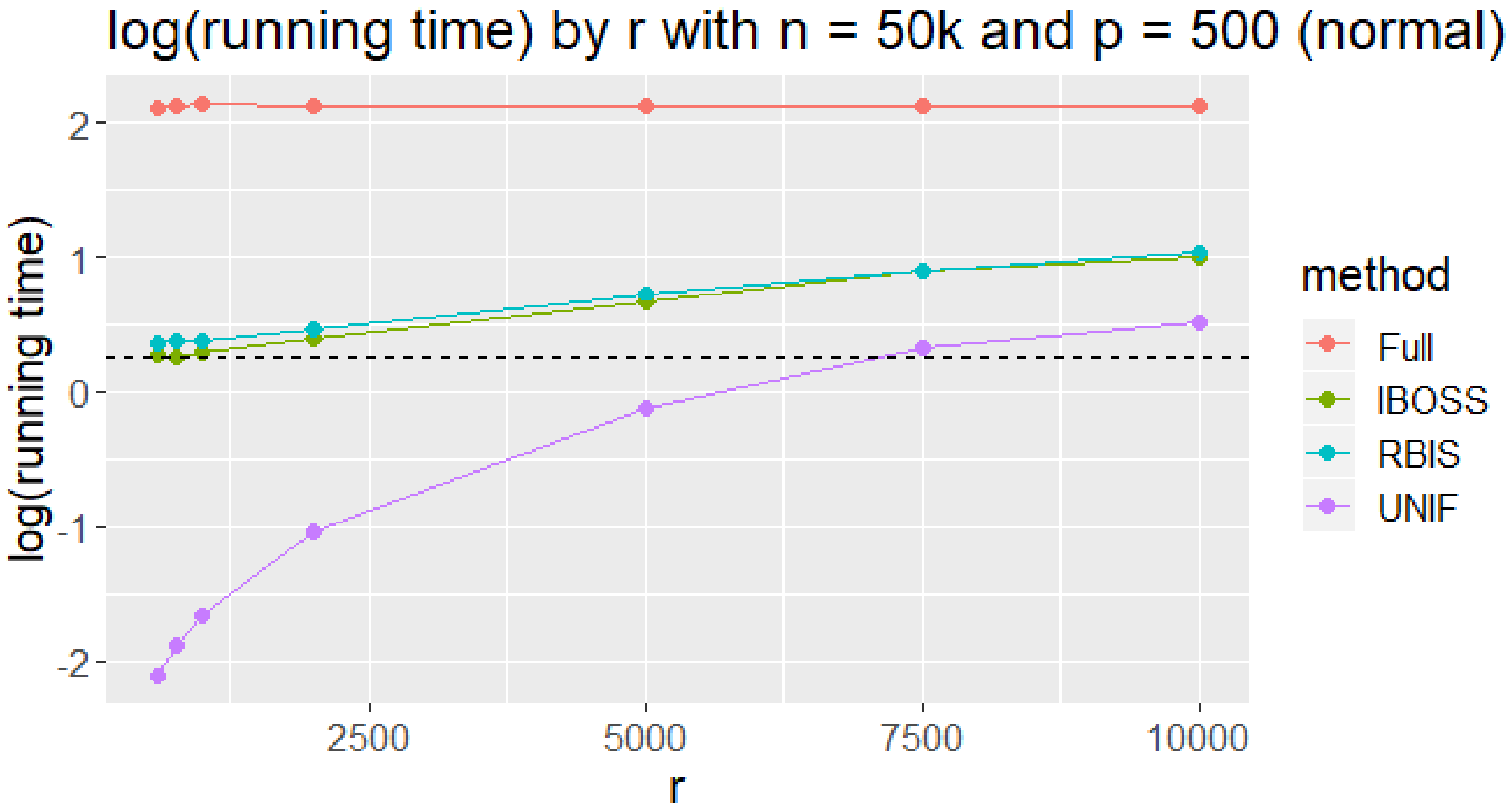

Figure: Time plot (promising algorithms only) with $n = 50k$ and $p = 500$

Simulation #2 fix n and p with varying r

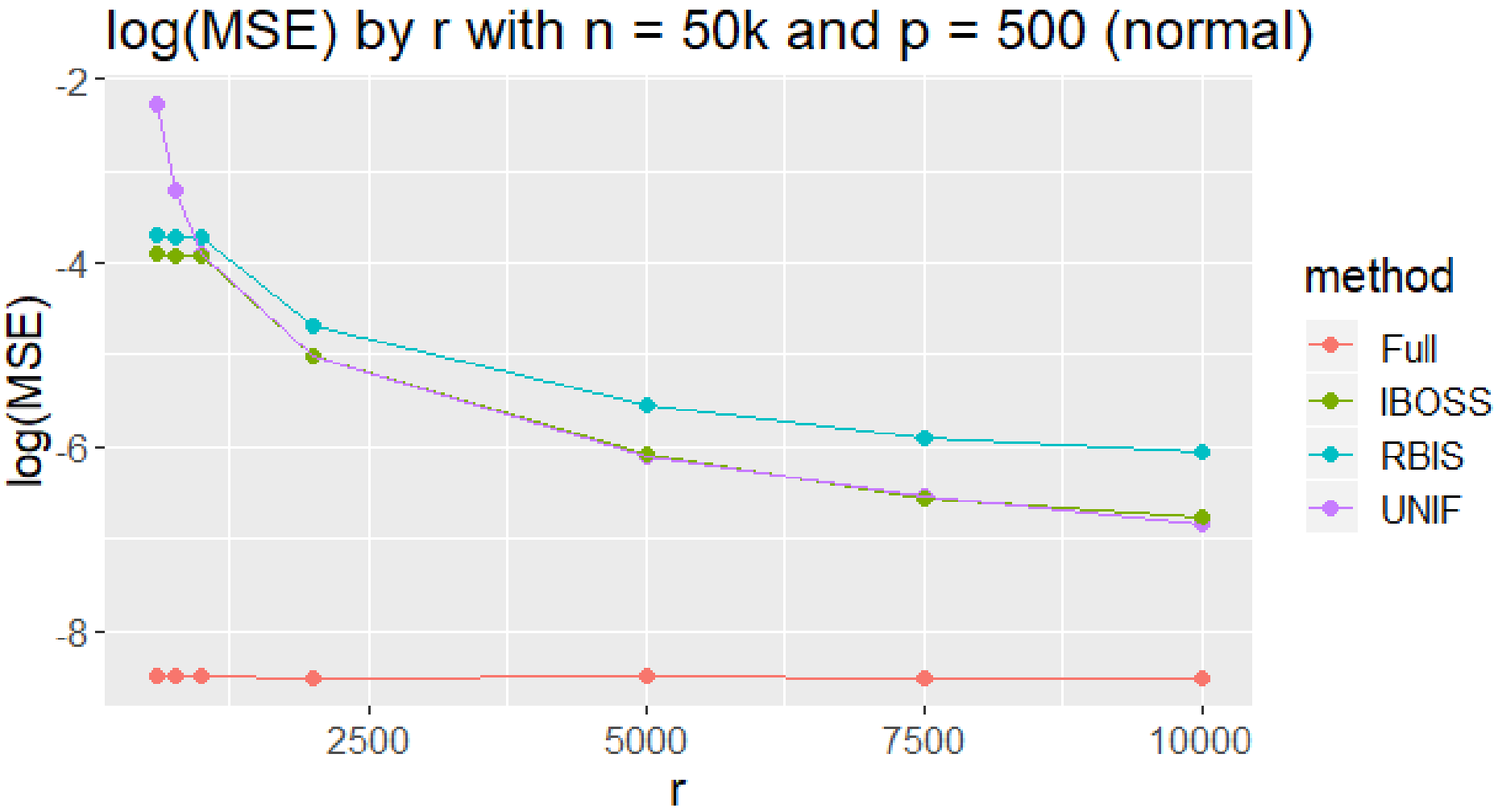

Figure: MSE plot (promising algorithms only) with $n = 50k$ and $p = 500$

Simulation #3 T_1 distribution (non-uniform leverages)

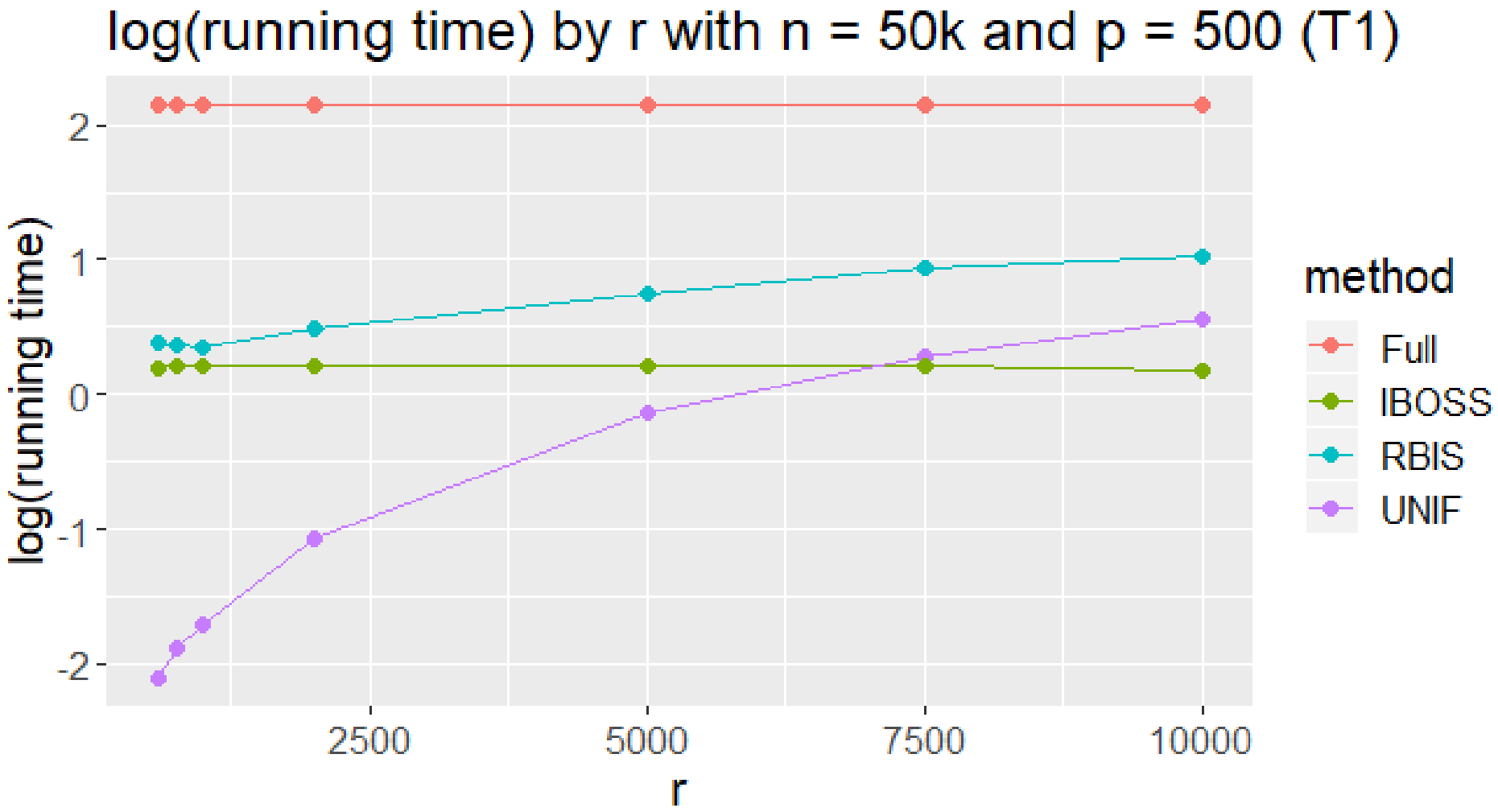

Figure: Time plot (T_1) with $n = 50k$ and $p = 500$

Simulation #3 T_1 distribution

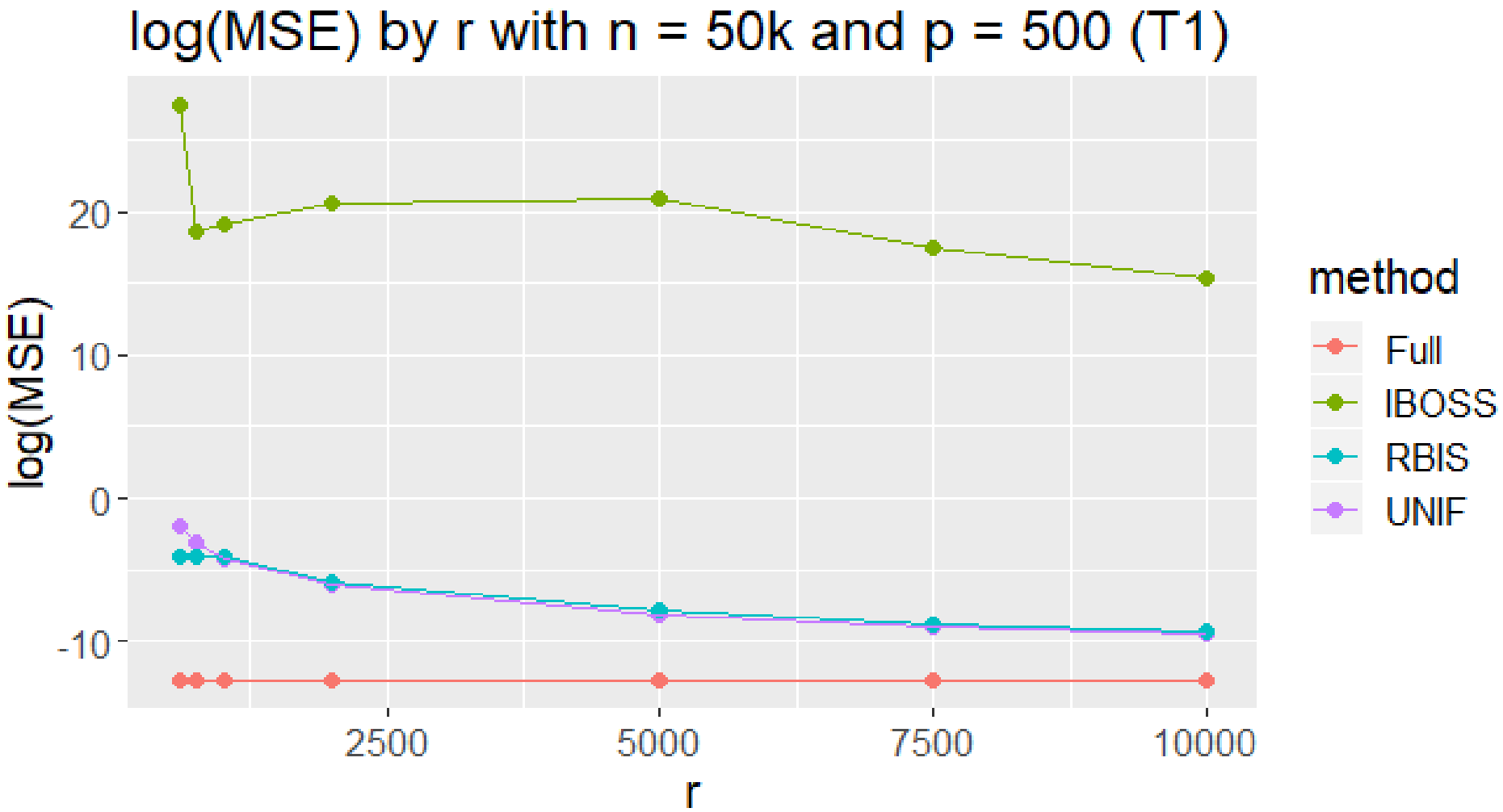

Figure: MSE plot (T_1) with $n = 50k$ and $p = 500$

Simulation #4 T_5 distribution

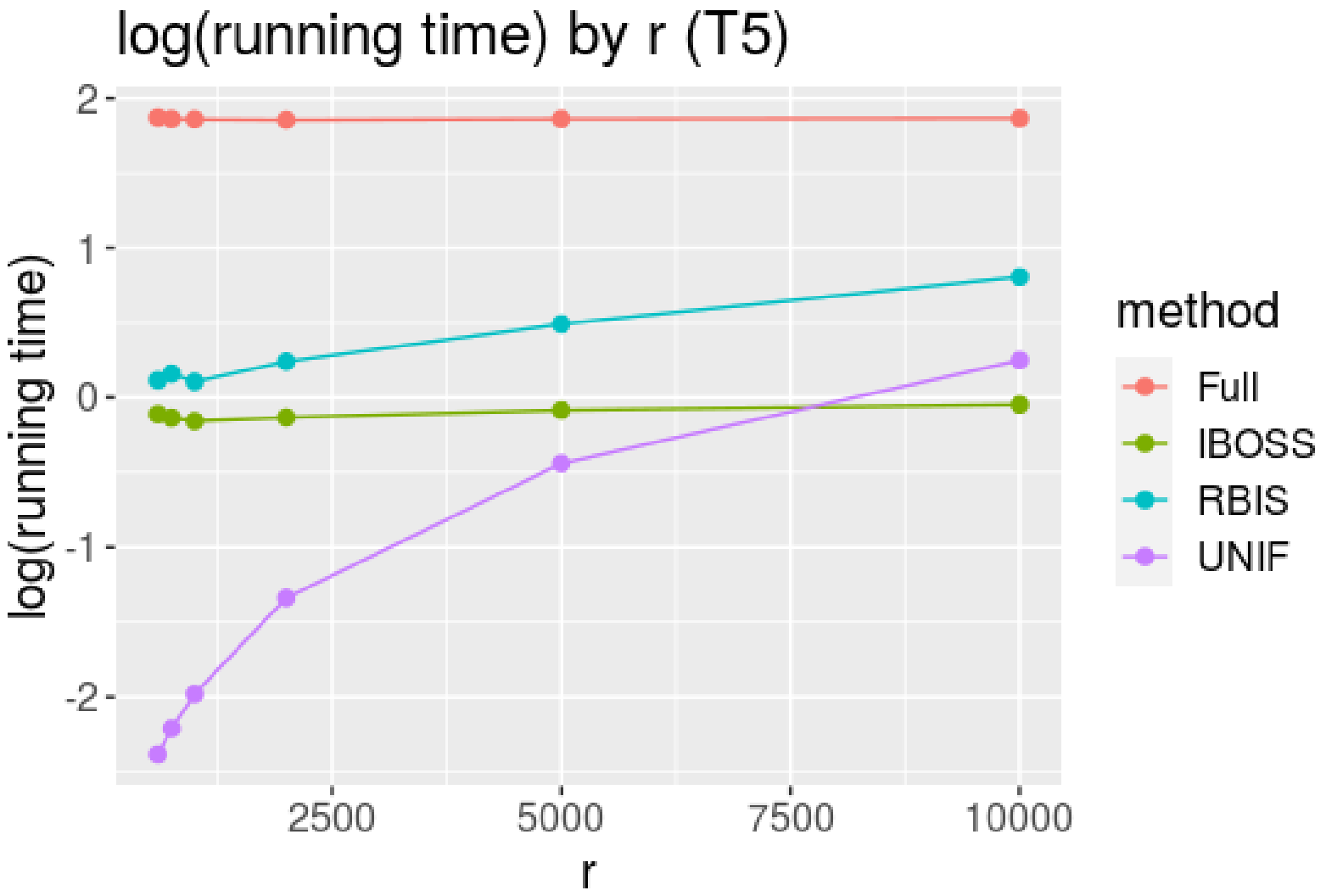

Figure: Time plot (T_5) with $n = 50k$ and $p = 500$

Simulation #4 T_5 distribution

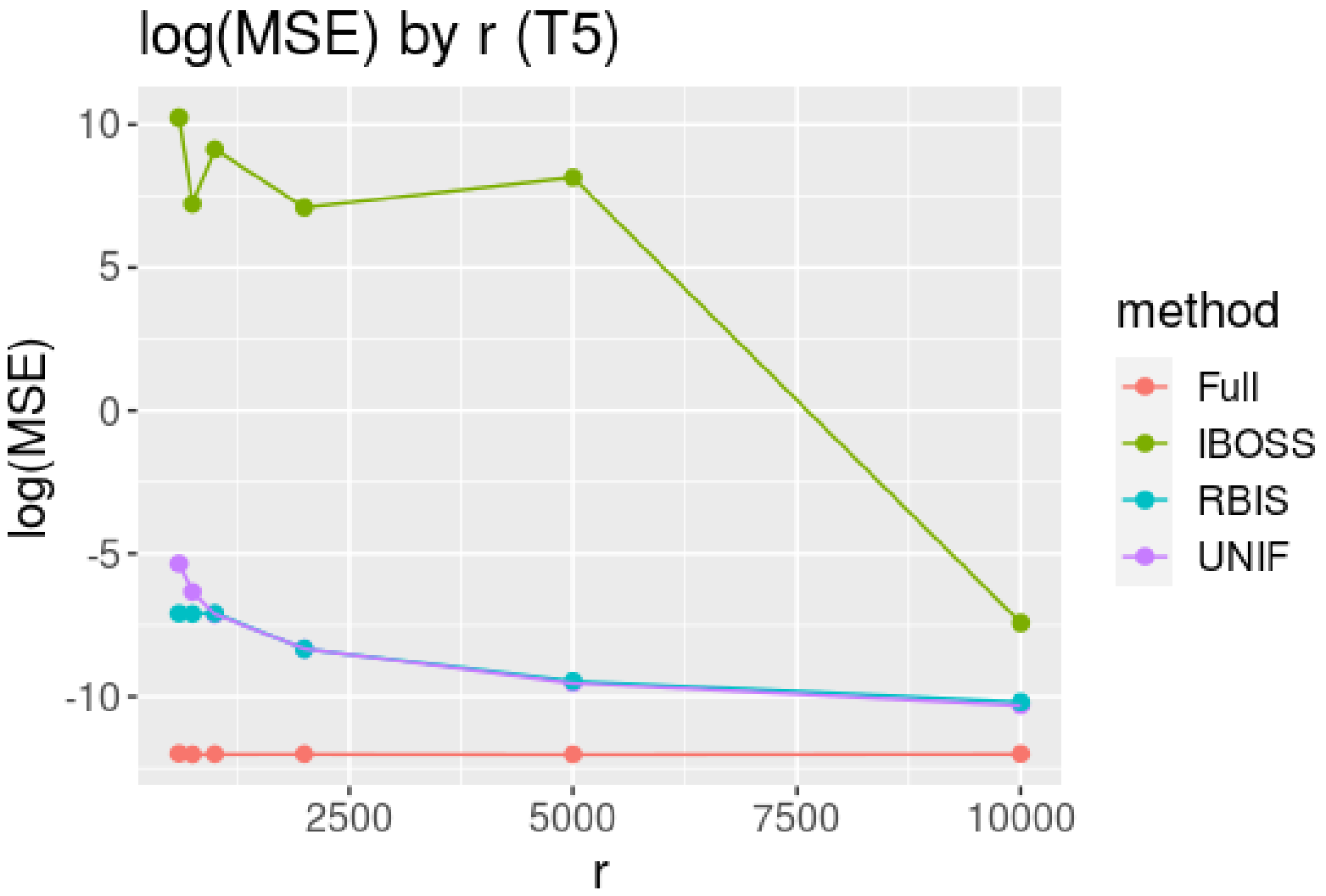

Figure: Time plot (T_5) with $n = 50k$ and $p = 500$

Discussions and Limitations of Methods

- Again, our researches and findings are still preliminary
- The variables here are restricted to continuous
- Some codings can definitely be improved
- It might be challenging to consider fitting interaction and quadratic terms using *IBOSS* and *RBIS*
- For uniform leverages, *IBOSS* seems to work just fine; for non-uniform leverages, *RBIS* and *UNIF* seems to work better

Conclusions and Possible Future Work

- From our findings, yet only *IBOSS* and *RBIS* have the potential to beat full OLS algorithm, not *BFSLEV*
- Future work includes
  - Theoretical justification of *RBIS*
  - Find a perfection of *RBIS* algorithm (finding its asymptotic bias/mse, and seeking to somehow remove its bias)
  - Seeking to find better coding (partial ordering) for *IBOSS* and *RBIS*
  - Consider higher-order terms like quadratic and interaction terms

References

P. Drineas, M. Mahoney and D. Woodruff (2012) Fast Approximation of Matrix Coherence and Statistical Leverage

Journal of Machine Learning Research **2012**, **13**, 3475–3506.

M. Kutner, C. Nachtsheim, J. Neter and W. Li (2004) Applied Linear Statistical Models

McGraw-Hill Irwin **2004**, **5e**.

P. Ma, M. Mahoney and B. Yu (2015) A Statistical Perspective on Algorithmic Leveraging

Journal of Machine Learning Research **2015**, **16**, 861–911.

Key References

P. Ma and X. Sun (2014) Leveraging for big data regression

WIREs Comput Stat **2014**.

C. Meng, Y. Wang, X. Zhang, A. Mandal and P. Ma (2017) Effective Statistical Methods for Big Data Analytics

Handbook of Research on Applied Cybernetics and Systems Science, IGI Global **2017**, 280–299.

H. Wang, M. Yang and J. Stufken (2019) Information-Based Optimal Subdata Selection for Big Data Linear Regression

Journal of American Statistical Association **2019**, **114:525**, 393–405.