Optimizing Bootstrapping Algorithm using R and Hadoop

Authors: Shicai Wang, Mihaela A. Mares and Yike Guo
Email: s.wang11@imperial.ac.uk

Presenter: Chen Feng

Imperial College London

Outline

- Background
- Design
- Evaluation

Imperial College London

Background

- In many machine learning application, the number of sample is relative small compared to the number of features.
- Bioinformatics
  - molecular variables (gene variants, protein abundance) vs individuals
- Resampling algorithms
  - E.g. Bolasso

Imperial College London

Background

- Bolasso Algorithm

Algorithm 1 Modified Bolasso

Input: data $(X_1, .., X_p, Y)$, for n samples
b number of bootstrap replicates
for $k = 1$ **to** b **do**
  Sample with replacement n new samples $(X_1^k, .., X_p^k, Y^k)$, from the input data
  Compute Lasso estimates β^k for best regularization λ identified on 100-fold cross-validation
  Compute vector $I^k = \{j|\beta_j \neq 0\}$
end for
for $i = 1$ **to** m **do**
  Compute $J^i = \{\beta^k|\beta_j \neq 0\, for\, at\, least\, i\%\, of\, the\, I^k\}$
end for
Compute final β using J^i with minimal cross validation error

Imperial College London

Background

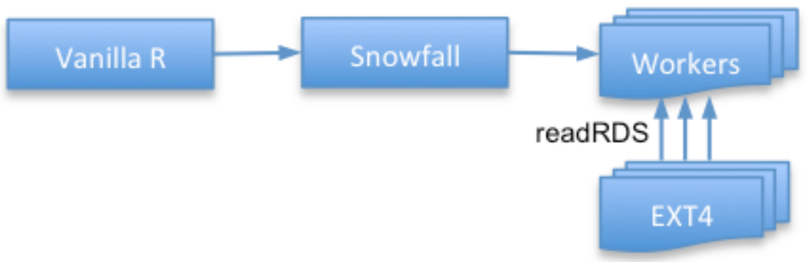
- Parallel R packages
  - RHIPE and RHadoop => MapReduce/Hadoop
  - SparkR and RABID => Spark
  - Snow and Snowfall
  - Parallel and doParallel

Imperial College London

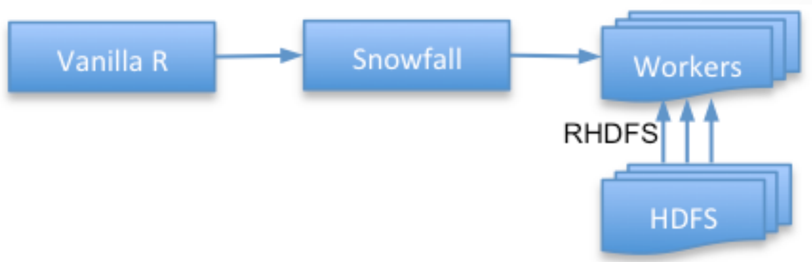
Background

- Parallel R approaches
  - Basic parallel R

  - Improved parallel R



  - Improved parallel R using HDFS



Imperial College London

Design

- Parallel Bolasso algorithm

Algorithm 2 Parallel Bolasso

Input: data $(X_1, .., X_p, Y)$, for n samples
b number of bootstrap replicates
for $k = 1$ **to** b **do**
  Sample with replacement n new samples $(X_1^k, .., X_p^k, Y^k)$ from the input data
  Store file for sample k on HDFS
end for
Call workers to compute results for all bootstraps $1..k$ in parallel
for $i = 1$ **to** m **do**
  Compute $J^i = \{\beta^k | \beta_j \neq 0\, for\, at\, least\, i\%\, of\, the\, I^k\}$
end for
Compute final β using J^i with minimal cross validation error

Design

- Bolasso worker function

Algorithm 3 worker function

Input: k as the bootstrap id
read $(X_1^k, .., X_p^k, Y^k)$ from HDFS
Compute Lasso estimates β^k for best regularization λ identified on 100-fold cross-validation
Compute vector $I^k = \{j|\beta_j \neq 0\}$
Return I^k

Imperial College London

Design

- Scheduler

Algorithm 4 Scheduling decision function

Input: j_n the number of jobs
$t(1), t(2), ..., t(m)$ execution times for the m VMs
Compute $times = \dfrac{max(t(1), t(2), ..., t(m))}{min(t(1), t(2), ..., t(m))}$
if $j_n / \sum(n(i)) > times$ **then**
  use the dynamic schedule (fine-grained case)
else
  use pre-schedule (coarse-grained case)
end if

Imperial College London

Design

- Dynamic schedule method
    - At the beginning, the scheduler initializes a 'first in first out queue' for all the jobs
    - Then, sends a job to each process and waits for a result return.
    - If any process finishes its job, the scheduler will send the next job in the queue to the process until the queue is empty.

Imperial College London

Design

- Pre-scheduled method
  - At the first round, each VM was assigned to the same number of workers and processes a similar workload. We summarized the execution time of all the tasks in m VMs, t(1), t(2),..., t(m).
  - Then, the balanced worker number in each VM, n(1), n(2),..., n(m), should follow the equation:
    n(1) : n(2) :...: n(m) = 1/t(1) : 1/t(2) :...: 1/t(m).

Imperial College London

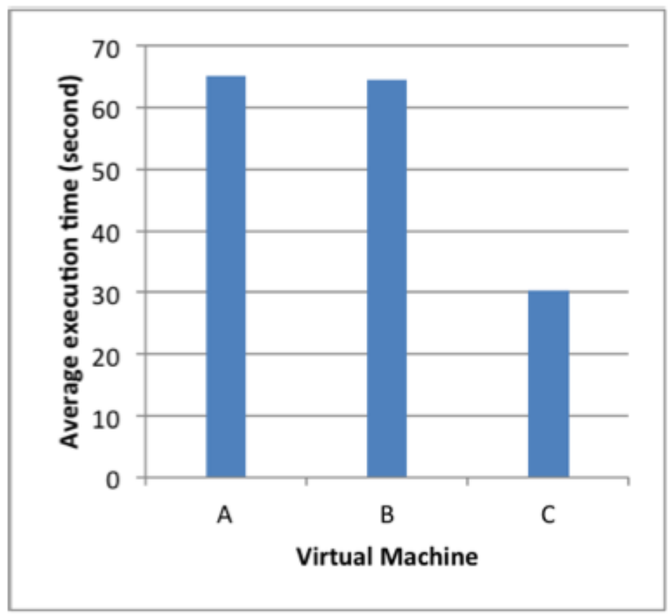
Evaluation

- Micro-benchmark
  - Environment (VM in OpenStack platform)
    - 6 VM (8 CPU, 8 GB memory), 3 VMs (A, B, C), each with 7 workers, are used for parallel R functions and the other 3 VMs (D, E, F) for HDFS system.
  - Datasets
    - 200 samples each with 10,000 features

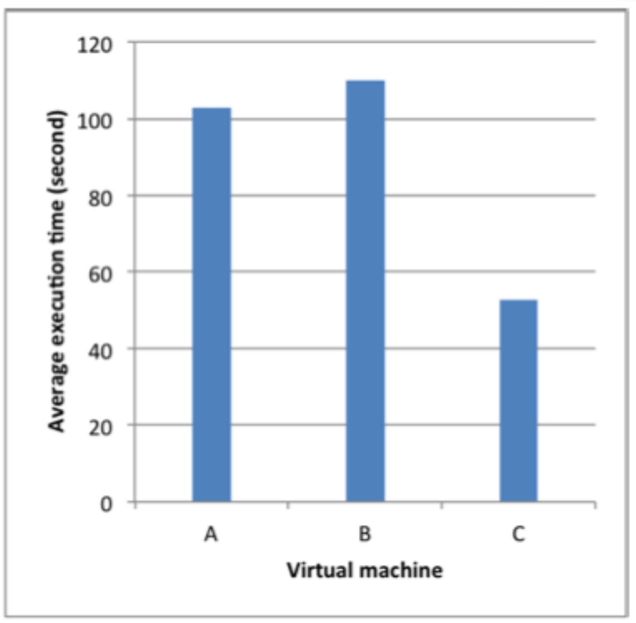
Imperial College London

Evaluation

- Micro-benchmark
  - Assessing the computation power of each machine



100 sample

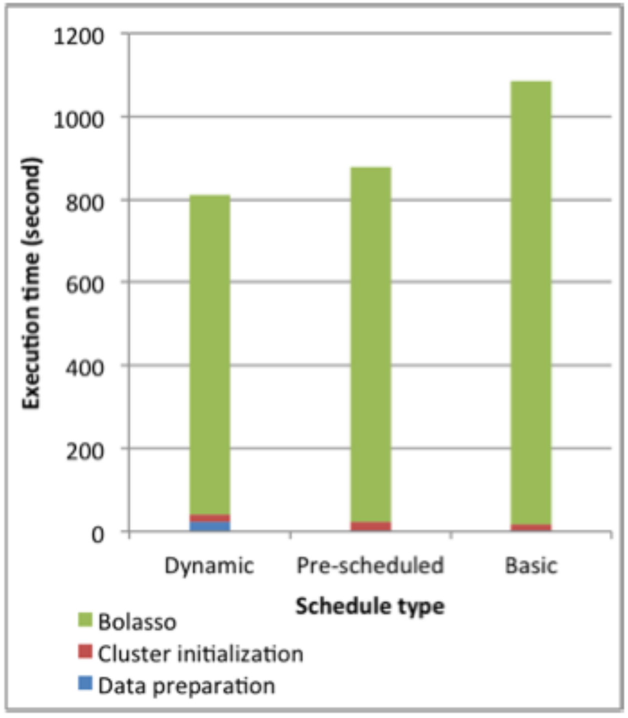


200 sample

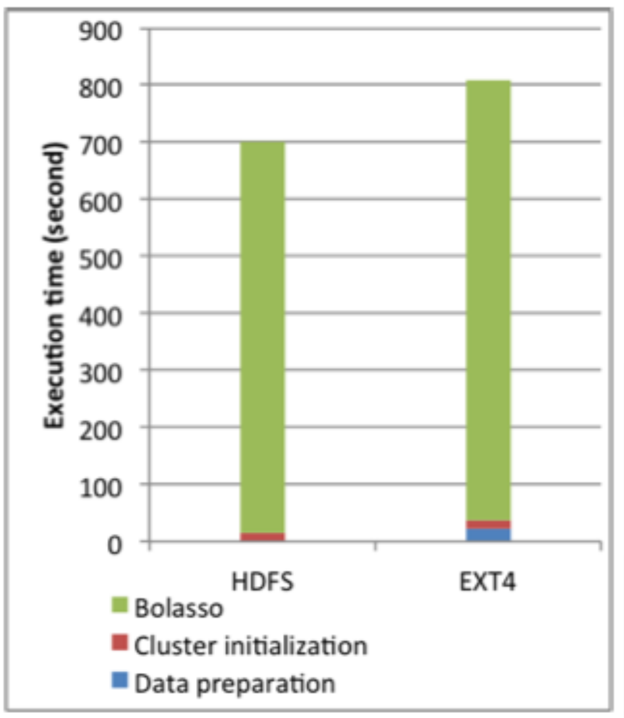
Imperial College London

Evaluation

- Micro-benchmark



Different scheduler
(EXT4)



HDFS v.s EXT4
(Dynamic)

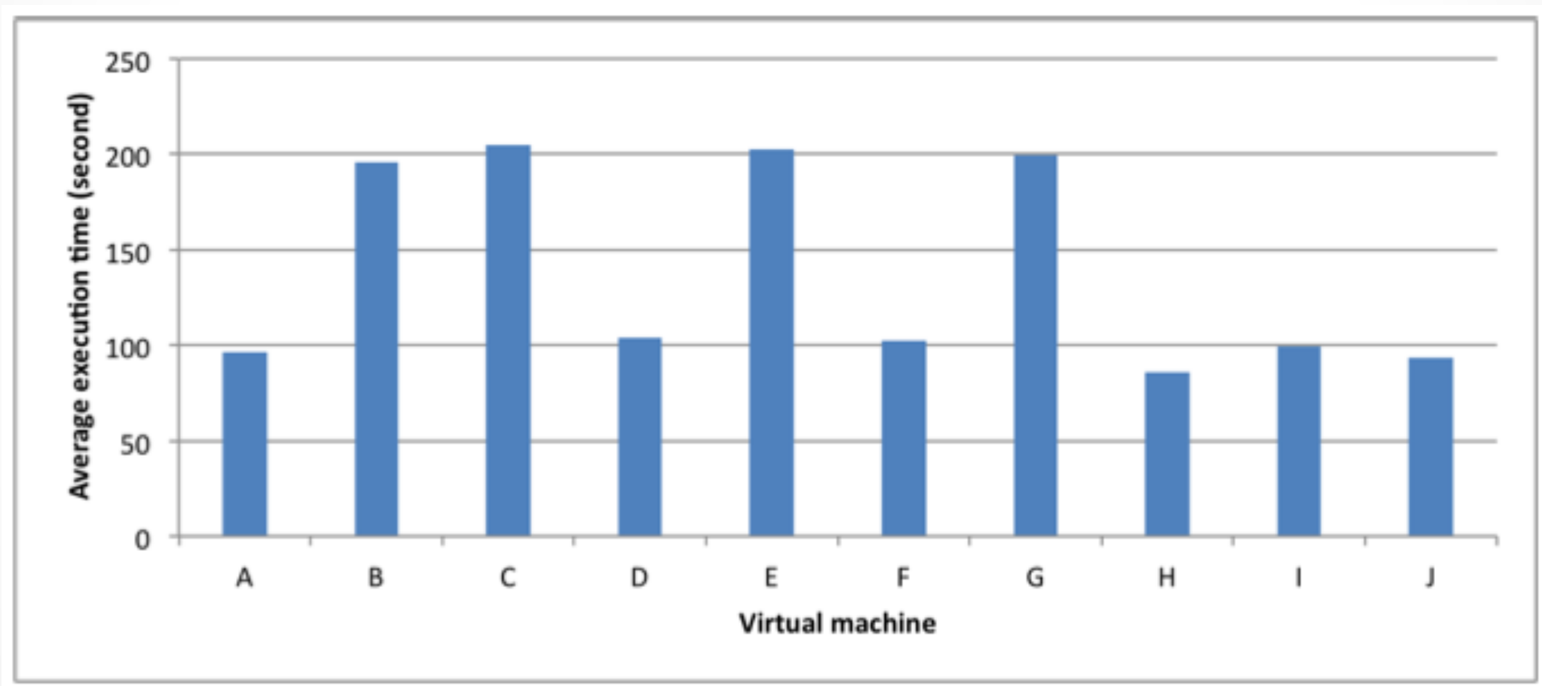
Evaluation

- Macro-benchmark
  - Environment (VM in OpenStack platform)
    - 14 VM (8 CPU, 8 GB memory), 10 VMs (A, B,…, J), each with 7 workers, are used for parallel R functions and the other 4 VMs (K, L, M, N) for HDFS system.
  - Datasets
    - 300 samples each with 10,000 features

Imperial College London

Evaluation

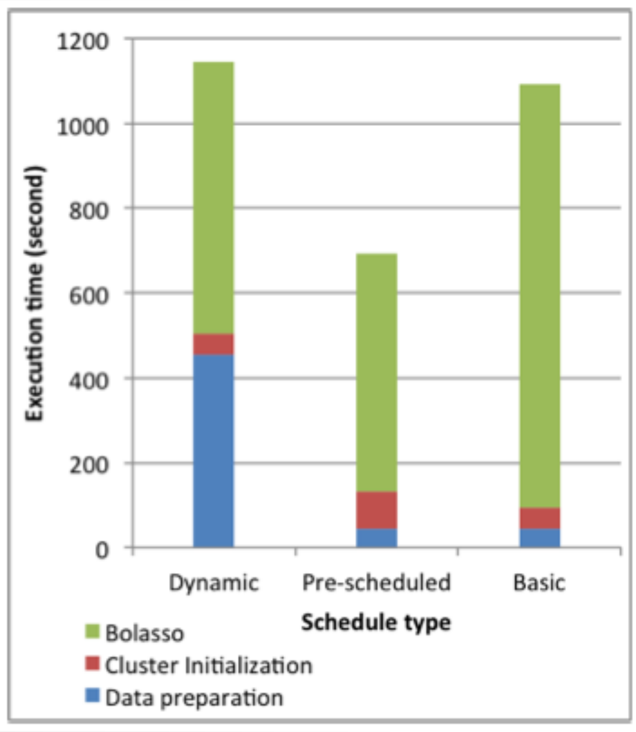
- Macro-benchmark
  - Assessing the computation power of each machine



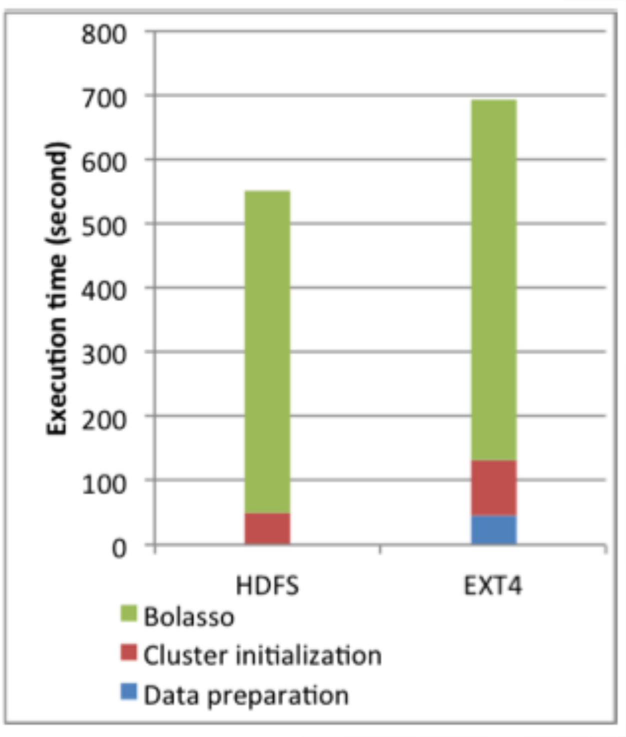
300 sample

Evaluation

- Macro-benchmark



Different scheduler
(EXT4)



HDFS v.s EXT4
(Pre-scheduled)

Conclusion

- The performance evaluation found that the new R on HDFS and its implementation in Snowfall and RHDFS saved up to half of the time than the conventional algorithm with Linux EXT4.

Imperial College London

Q & A

Thanks!

Imperial College London