A Single Program Multiple Data Algorithm for Feature Selection

Paper 153

Intelligent Systems Design and Applications (ISDA 2018)

December 6 - 8 2018, VIT Vellore, India

Bhabesh Chanduka, Tushaar Gangavarapu, Jaidhar C D

bhabesh.chanduka@gmail.com tushaargvsg45@gmail.com jaidharcd@nitk.edu.in

Dept. of Information Technology, National Institute of Technology Karnataka

Please cite this article as: B. Chanduka, T. Gangavarapu, and C.D. Jaidhar, "A Single Program Multiple Data Algorithm for Feature Selection," International Conference on Intelligent Systems Design and Applications. Springer, Cham, 2018.

Agenda

- Introduction
- mRMR Feature Selection Technique
- A Data Parallel Approach to mRMR Feature Selection
- Results
- Conclusion

Introduction

Feature Selection is the process of selecting a subset of crucial features from a given set of features, for efficient model construction.

Below are few types of feature selection techniques:

- ❏ Filter
- ❏ Wrapper
- ❏ Embedded

mRMR Feature Selection Techniques

The mRMR (**m**inimum **R**edundancy **M**aximum **R**elevance) procedure, proposed by Peng et al. [1], is an algorithm to perform feature selection by trading-off between *relevance* and *redundancy* (by taking the ratio).

Below are few earlier approaches to improve mRMR:

- ❏ Distributed version using Scala [Gellego et al., 2017]
- ❏ MapReduce [Reggiani et al., 2017]
- ❏ Artificial Bee Colony [Alshamlan et al., 2015]
- ❏ Task parallel approach [LeKhac et al., 2013]

A Data Parallel Approach to mRMR Feature Selection

To overcome the dependency on distributed systems for effective modeling and subset selection, we propose a **data parallel approach** for mRMR feature selection.

- Initialize an empty set denoting the set of features selected
- Pre-calculate the relevance of each feature with the class (can be done in parallel in the number of features is large)
- Select the feature that has highest relevance with the class
- Calculate in parallel the redundancy of each feature with this newly added feature

A Data Parallel Approach to mRMR Feature Selection
(contd.)

- Select the feature that has the highest relevance with the class and minimum redundancy with the selected feature (done in parallel)
- Repeat the above step in parallel to select the features
- When a feature is selected, calculate redundancies of all other features with this feature

If N denotes the total number of features in the dataset:

Space Complexity : $O(N^2)$ Time Complexity : $O(N^2)$

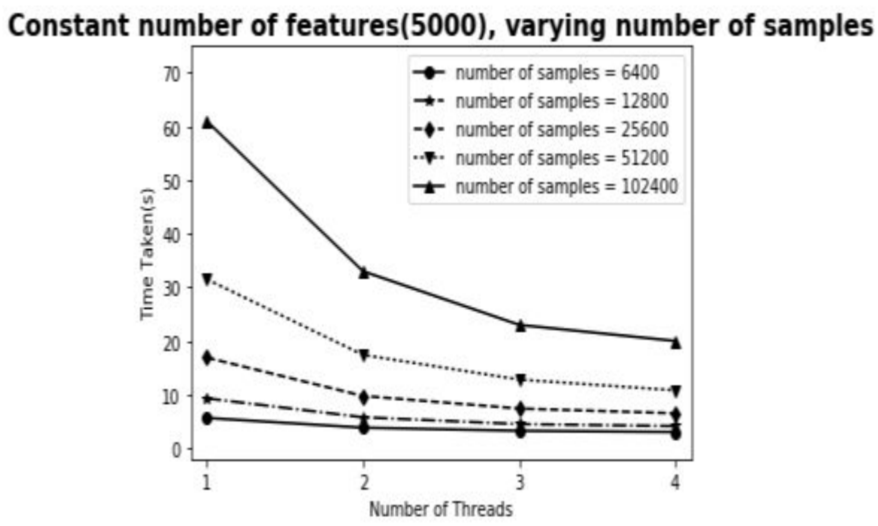
Results

Speed-up: Ratio of sequential execution time and parallel execution time

$$S = \frac{T_{sequential}}{T_{parallel}}$$

Performance Scalability on Number of Samples

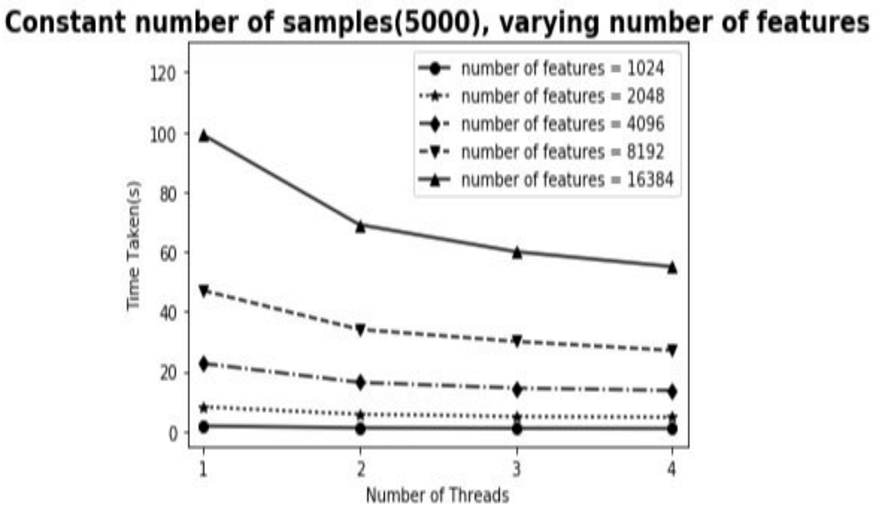
With increasing number of samples, the speed-up obtained with increasing number of threads is more



Results (contd.)

Performance Scalability on Number of Features

With increasing number of features, the speed-up obtained with increasing number of threads is more



Conclusion

- The time complexity has been reduced from cubic to quadratic
- The effect of parallelization is more pronounced for larger, higher dimensional datasets

In the future, approaches to reduce the space complexity will be explored

References

[1] H. Peng, F. Long and C. Ding, ”Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy”, in IEEE Transaction on Pattern Analysis and Machine Intelligence.

[2] Ramrez-Gallego, Sergio et al. Fast-mRMR: Fast Minimum Redundancy Maximum Relevance Algorithm for High-Dimensional Big Data. Int. J. Intell. Syst. 32 (2017): 134-152.

[3] Reggiani, Claudio et al. Feature selection in high-dimensional dataset using MapReduce., BNCAI (2017).

[4] Alshamlan H, Badr G, Alohali Y. mRMR-ABC: A Hybrid Gene Selection Algorithm for Cancer Classification Using Microarray Gene Expression Profiling. Biomed Res Int. 2015;2015:604910.

[5] LeKhac N., Wu B., Chen C., Kechadi MT. (2013) Feature Selection Parallel Technique for Remotely Sensed Imagery Classification. In Computational Science and Its Applications ICCSA 2013. Lecture Notes in Computer Science, vol 7972. Springer, Berlin, Heidelber.