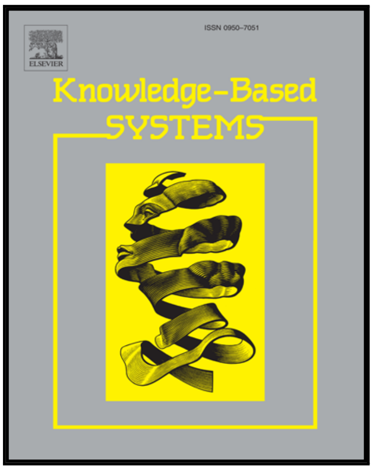

kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors Classifier for Big Data

Jesus Maillo, Sergio Ramírez, Isaac Triguero, Francisco Herrera

PII: S0950-7051(16)30175-7
DOI: 10.1016/j.knosys.2016.06.012
Reference: KNOSYS 3567

To appear in: *Knowledge-Based Systems*

Received date: 31 January 2016
Revised date: 10 June 2016
Accepted date: 12 June 2016

Please cite this article as: Jesus Maillo, Sergio Ramírez, Isaac Triguero, Francisco Herrera, kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors Classifier for Big Data, *Knowledge-Based Systems* (2016), doi: 10.1016/j.knosys.2016.06.012

kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors Classifier for Big Data

Jesus Maillo[a,*], Sergio Ramírez[a], Isaac Triguero[c,d,e], Francisco Herrera[a,b]

[a] *Department of Computer Science and Artificial Intelligence, University of Granada, CITIC-UGR, Granada, Spain, 18071*

[b] *Faculty of Computing and Information Technology, University of Jeddah, Jeddah, Saudi Arabia, 21589*

[c] *Department of Internal Medicine, Ghent University, Ghent, Belgium, 9000*

[d] *Data Mining and Modelling for Biomedicine group, VIB Inflammation Research Center, Zwijnaarde, Belgium, 9052*

[e] *School of Computer Science, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham NG8 1BB, United Kingdom*

Abstract

The k-Nearest Neighbors classifier is a simple yet effective widely renowned method in data mining. The actual application of this model in the big data domain is not feasible due to time and memory restrictions. Several distributed alternatives based on MapReduce have been proposed to enable this method to handle large-scale data. However, their performance can be further improved with new designs that fit with newly arising technologies.

In this work we provide a new solution to perform an exact k-nearest neighbor classification based on Spark. We take advantage of its in-memory operations to classify big amounts of unseen cases against a big training dataset. The map phase computes the k-nearest neighbors in different training data splits. Afterwards, multiple reducers process the definitive neighbors from the list obtained in the map phase. The key point of this proposal lies on the management of the test set, keeping it in memory when possible. Otherwise, it is split into a minimum number of pieces, applying a MapReduce per chunk, using the caching skills of Spark to reuse the previously partitioned

[*] Corresponding author. Tel : +34 958 240598; Fax: + 34 958 243317
Email addresses: `jesusmh@decsai.ugr.es` (Jesus Maillo), `sramirez@decsai.ugr.es` (Sergio Ramírez), `Isaac.Triguero@nottingham.ac.uk` (Isaac Triguero), `herrera@decsai.ugr.es` (Francisco Herrera)