Accepted Manuscript

Efficient multiple incremental computation for Kernel Ridge Regression with Bayesian uncertainty modeling

Bo-Wei Chen, Nik Nailah Binti Abdullah, Sangoh Park, Y. Gu

PII: S0167-739X(17)30811-7
DOI: https://doi.org/10.1016/j.future.2017.08.053
Reference: FUTURE 3648

To appear in: *Future Generation Computer Systems*

Received date : 29 April 2017
Revised date : 5 August 2017
Accepted date : 28 August 2017

Please cite this article as: B.-W. Chen, N.N.B. Abdullah, S. Park, Y. Gu, Efficient multiple incremental computation for Kernel Ridge Regression with Bayesian uncertainty modeling, *Future Generation Computer Systems* (2017), https://doi.org/10.1016/j.future.2017.08.053

Efficient Multiple Incremental Computation for Kernel Ridge Regression with Bayesian Uncertainty Modeling

Bo-Wei Chen, Nik Nailah Binti Abdullah, Sangoh Park, and Y. Gu

***Abstract*—This study presents an efficient incremental/decremental approach for big streams based on Kernel Ridge Regression (KRR), a frequently used data analysis in cloud centers. To avoid reanalyzing the whole dataset when sensors receive new training data every time, typical incremental KRR used a single-instance mechanism for updating an existing system. However, this inevitably increased redundant computational time, not to mention applicability to big streams. To this end, the proposed mechanism supports incremental/decremental processing for both single and multiple samples (i.e., batch processing). A large scale of data can be divided into batches, processed by a machine, without sacrificing the accuracy. Moreover, incremental/decremental analyses in empirical and intrinsic space are also proposed in this study to handle different types of data with a large number of samples or feature dimensions, whereas typical methods focused only on one type. At the end of this study, we further the proposed mechanism to statistical Kernelized Bayesian Regression, so that uncertainty modeling with incremental/decremental computation becomes applicable. Experimental results showed that computational time was significantly reduced, better than the original nonincremental design and the typical single incremental method. Furthermore, the accuracy of the proposed method remained the same as the baselines. This implied that the system enhanced efficiency without sacrificing the accuracy. These findings proved that the proposed method was appropriate for variable streaming data analysis, thereby demonstrating the effectiveness of the proposed method.**

***Index Terms* — Multiple incremental analysis, multiple decremental analysis, incremental learning, kernel ridge regression (KRR), recursive KRR, uncertainty analysis, kernelized Bayesian regression, Gaussian process, batch learning, online learning, regression, classification**

I. Introduction

Ridge regression extends linear regression techniques, where a ridge parameter is imposed on the objective function to regularize and prevent a model [1] from overfitting. Such regularization uses Λ_2 norm, or Euclidean distance, as the criterion for constraining the searching path of objective functions. Kernel Ridge Regression (KRR) further advances ridge regression by mapping feature space into hyperspace with the use of kernel functions, for example, polynomial functions and Radial Basis Functions (RBFs). In machine learning research, KRR and Support Vector Machines (SVMs) have been widely used in pattern classification, especially in recent decentralized wireless sensor networks and computing platforms for the Internet of Things (IoTs).

Although KRR has a closed-form solution, which involves the inverse of matrices, calculating these matrices degrades computational speeds [2]. Literature reviews [1] showed that the complexity of KRR [3] was as high as O(N^3), whereas that of SVMs was O(N^2), in which N stands for the number of instances in data. Such a characteristic is a burden in cloud servers, which consume too much power for computation, not to mention online streaming data analysis for IoTs [4]. The source nodes can rapidly collect information and transmit it to a fusion center, or a sink node [5, 6], which is designed for data pooling [7]. The massive amount of streams may deplete computational resources. This requires either distributed processing [8, 9] or incremental analysis [10] to deal with big streams.

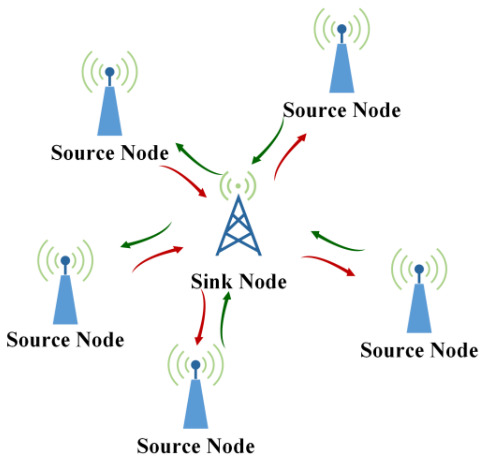

Fig. 1. Data pooling in sink nodes for wireless sensor networks

Unlike distributed processing, incremental analysis allows the system to add new training samples and to update itself without rescanning and reanalyzing existing datasets [11]. This is because incremental algorithms can reserve earlier

B.-W. Chen and N. N. B. Abdullah are with the School of Information Technology, Monash University, Australia
S. Park is with the School of Computer Engineering, Chungang University, South Korea
Y. Gu is with the University of Science and Technology Beijing and also with Beijing University of Chemical Technology, China.