Applied Mathematics and Computation 275 (2016) 411–421

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Large-scale linear regression: Development of high-performance routines

Alvaro Frank, Diego Fabregat-Traver*, Paolo Bientinesi

RWTH Aachen University, AICES, Schinkelstr. 2, 52062 Aachen, Germany

ARTICLE INFO

Keywords:
Linear regression
Ordinary least squares
Algorithm design
Out-of-core
Parallelism
Scalability

ABSTRACT

In statistics, series of ordinary least squares problems (OLS) are used to study the linear correlation among sets of variables of interest; in many studies, the number of such variables is at least in the millions, and the corresponding datasets occupy terabytes of disk space. As the availability of large-scale datasets increases regularly, so does the challenge in dealing with them. Indeed, traditional solvers—which rely on the use of "black-box" routines optimized for one single OLS—are highly inefficient and fail to provide a viable solution for big-data analyses. As a case study, in this paper we consider a linear regression consisting of two-dimensional grids of related OLS problems that arise in the context of genome-wide association analyses, and give a careful walkthrough for the development of OLS-GRID, a high-performance routine for shared-memory architectures; analogous steps are relevant for tailoring OLS solvers to other applications. In particular, we first illustrate the design of efficient algorithms that exploit the structure of the OLS problems and eliminate redundant computations; then, we show how to effectively deal with datasets that do not fit in main memory; finally, we discuss how to cast the computation in terms of efficient kernels and how to achieve scalability. Importantly, each design decision along the way is justified by simple performance models. OLS-GRID enables the solution of 10^{11} correlated OLS problems operating on terabytes of data in a matter of hours.

1. Introduction

Linear regression is an extremely common statistical tool for modeling the relationship between two sets of data. Specifically, given a set of "independent (scalar) variables" $x_{h1}, x_{h2}, \ldots, x_{hp}$, and a "dependent (scalar) variable" y_h, one seeks the correlation terms β_k, $k = 1, \ldots, p$, in the linear model

$$y_h = \beta_1 x_{h1} + \cdots + \beta_p x_{hp} + \epsilon_h, \qquad h = 1, \ldots, n. \tag{1}$$

In matrix form, Eq. (1) is expressed as $y = X\beta + \epsilon$, where $y \in R^n$ is a vector of n "observations", the columns of $X \in R^{n \times p}$ are "predictors" or "covariates", the vector $\beta = [\beta_1, \ldots, \beta_p]^T$ contains the "regression coefficients", and $\epsilon \in R^n$ is an error term that one wishes to minimize. In many disciplines, linear regression is used to quantify the relationship between one or more y's from the set $\mathcal{Y}$, and each of many x's from the set $\mathcal{X}$. The computational challenges raise from the all-to-all nature of the problem (estimate how strongly each of the covariates is related to each of the observations), and from the sheer size of the datasets $\mathcal{Y}$ and $\mathcal{X}$, which often cannot be stored directly in main memory.

* Corresponding author. Tel.: +49 2418099128.
E-mail addresses: alvaro.frank@rwth-aachen.de (A. Frank), fabregat@aices.rwth-aachen.de (D. Fabregat-Traver), pauldj@aices.rwth-aachen.de (P. Bientinesi).

http://dx.doi.org/10.1016/j.amc.2015.11.078
0096-3003/

One of the standard approaches to fit the model (1) to given y and X is by solving an ordinary least squares (OLS) problem; in linear algebra terms, this corresponds to computing the vector β such that

$$\beta = \left(X^T X\right)^{-1} X^T y.$$

In typical datasets, $\hat{m}$, the number of available covariates ($\hat{m} = |\mathcal{X}|$), is much larger than p, the number of variables actually used in the model. In this case, a group of $l < p$ covariates is kept fixed, and the remaining $p - l$ slots are filled from $\mathcal{X}$, in a rotating fashion; it is not uncommon that the value $p - l$ is very small, often just one, thus leading to $m \geq \hat{m}$ distinct OLS problems. Mathematically, this means computing a series of β_i's such that

$$\beta_i = \left(X_i^T X_i\right)^{-1} X_i^T y, \quad \text{where} \quad i = 1, \ldots, m; \tag{2}$$

here X_i consists of two parts: X_L, which contains l columns and is fixed across all m OLS problems, and X_{R_i}, which instead contains $p - l$ columns taken from $\mathcal{X}$. In many applications, m can be of the order of millions or even more.

When $t > 1$ dependent variables ($t = |\mathcal{Y}|$) are to be studied against $\mathcal{X}$, the problem (2) assumes the more general form

$$\beta_{ij} = \left(X_i^T X_i\right)^{-1} X_i^T y_j, \tag{3}$$

where $i = 1, \ldots, m$, and $j = 1, \ldots, t$, indicating that one has to compute a two-dimensional grid of β_{ij}'s, each one corresponding to an OLS problem. This is for instance the case in genomics (multi-trait genome-wide association analyses) [1] and econometrics (explanatory variable exploration) [2].

Despite the fact that OLS solvers are provided by many libraries and languages (e.g., LAPACK, NAG, MKL, Matlab, R), no matter how optimized those are, any approach that aims at computing the 2D grid (3) via $t \times m$ invocations of a "black-box" routine is entirely unfeasible. The main limitations come from the fact that this approach leads to the execution of inefficient and redundant operations, lacks a mechanism to effectively manage data transfers from and to hard disk, and underutilizes the resources on parallel architectures.

In this paper, we consider an instance of Eq. (3) as it arises in genomics, and develop OLS-GRID, a parallel solver tailored for this application. Specifically, we focus on the study of *omics* data[1] in the context of genome wide association analyses (GWAA).[2] Omics GWAA study the relation between m groups of genetic markers and t phenotypic traits in populations of n individuals. In terms of OLS, each trait is represented by a vector y_j containing the trait measurements (one per individual); each matrix $X_i = [X_L | X_{R_i}]$ is composed of a set of l fixed covariates such as sex, age, and height (X_L), and one of the groups of $r = p - l$ markers (X_{R_i}). A positive correlation between markers X_{R_i} and trait y_j indicates that the markers may have an impact in the expression of the trait.

Typical problem sizes in omics GWAA are roughly $10^3 \leq n \leq 10^5$, $2 \leq p \leq 20$ (with $r = 1$ or 2), $10^6 \leq m \leq 10^8$, and $10^2 \leq t \leq 10^5$. An exemplary analysis with size $n = 30{,}000$, $p = 10$ ($l = 8$, $r = 2$), $m = 10^7$, and $t = 10^4$, poses three considerable challenges. First, it requires the computation of 10^{11} OLS problems, which, if tackled by a traditional "black-box" solver, would perform $O(10^{18})$ floating point operations (flops). Despite the fact that the problem lends itself to a lower computational cost and efficient solutions, a black-box solver ignores the structure of the problem and requires large clusters to obtain a solution. The second challenge is posed by the size of the datasets to be processed: assuming single-precision data (4 bytes per element), a GWAA solver reads as input about 2.4TBs of data and produces as output 4TBs of data. If the data movement is not analyzed properly, the time spent in I/O transfers might render the computation unfeasible. Finally, the computation needs to be parallelized and organized so that the potential of the current multi-core and many-core CPUs is fully exploited.

Contributions. This paper is concerned with the design and the implementation of OLS-GRID, a high-performance algorithm for large-scale linear regression. While we use omics GWAA as a case study, the discussion is relevant to a range of OLS-based applications. Specifically, (1) we illustrate how to take advantage of the specific structure in the grid of OLS problems to design specialized algorithms, (2) analyze the data transfers from and to disk to effectively deal with large datasets that do not fit in main memory, and (3) discuss how to cast the computation in terms of efficient kernels and how to attain scalability on multi-core and many-core CPUs. Moreover, by making use of simple performance models, we identify the performance bottlenecks with respect to the problem size. OLS-GRID, available as part of the GenABEL suite [6], allows one to execute an analysis of the aforementioned size in less than 7 h on a 40-core node.

Related work. Genome-wide association analyses received a lot of attention in the last decade [7]. Numerous high-impact findings have been reported, including but not limited to the identification of genetic variations associated to a common form of blindness, type 2 diabetes, and Parkinson's disease [8–11]. A popular approach to GWAA is the so called Variance Components model, which boils down to a set of equations similar to Eq. (3). The main difference with the present work lies on the core equation, where one has to solve grids of generalized least squares (GLS) problems instead of grids of OLSs. A number of libraries have been developed to carry out GLS-based GWAA, the most relevant being FaST-LMM, GEMMA, GWFGLS, and OmicABEL [5,6,12,13].

[1] With the term *omics* we refer to large-scale analyses involving at least hundreds of traits [3–5].

[2] GWAA are often also referred to as genome wide association studies (GWAS) and whole genome association studies (WGAS).