COURSE PROJECT: BOUNDING LEAST SQUARES PROBLEM SOLUTIONS BASED ON INTERVAL ARITHMETICS

CARL NETTELBLAD

Background

In analysis of Quantitative Trait Loci (QTL), a specific technique in quantitative genetics, it is common to try to fit a linear model to data. In essence, what is done is to answer the question "how well can we model the values we see for some trait in some individuals, given the genetic data in a specific set of positions (loci) in the genome". The trait studied could be any scalar, such as usable mass of meat (in chicken), milk production (in cattle), or blood pressure (in humans). To do this, first experimental and computational methods need to be used to establish an estimate of the genetic content in each position.

After this follows a global optimization search, where repeated model fittings are done. If multiple genes are supposed to be interacting, pairs, triples etc of genetic positions need to be tested. This results in a rather bad scaling, where millions or billions of model fittings could be needed. For mostly practical reasons, most actual experimentalists stop at testing pairs.

Description

In related work, we have shown how opportunistic optimization algorithms can find the proper location quickly, due to the fact that genetic positions close together are also generally inherited together [1]. More recently, we could show a slight variation on this algorithm (PruneDIRECT [2]) with theoretical underpinnings in specific cases, which made it possible to make stronger claims that a minimum found using these techniques is in fact the correct one, even though not all positions have been tested.

However, that analysis relied on a specific problem structure. Unpublished results show that in some cases, it is possible to derive pessimistic analytical bounds on the possible residual square sum of a least squares regression over a *range* of genetic positions, based on the total known variation in each indicator variable in that range. This basically amounts to changing the A matrix in a linear equation system $Ax = b$ from a matrix of scalars to a matrix of intervals. If the bound shows that the full range cannot contain a value that is superior to the minimum found in a single point, the whole range can be excluded. Thus, the search can be focused on regions of relevance without ever excluding correct solutions.

This bound still assumes a specific model formulation, which is not the most common one used in actual experiments. We want to explore using general interval arithmetics techniques to apply this bounding approach to the QTL search problems for linear models.

Further details

The intent in this project is to attempt using existing libraries for solving linear interval equation systems, or (almost equivalently) finding the inverse for interval matrices. In order to accomplish the least squares fitting, we will start with the simplest possible approach, i.e. solving the normal equations, using the resulting solution interval vector to determine bounds on the residual sum of squares. Even if the bounds turn out to be too lose to be meaningful in this context, performance and accuracy comparisons can be done between interval-based and traditional solvers.

The main library intended for testing is ALIAS-C++ [1].

Aims

The project consists of several steps, where each has tangible benefits:

- Solving small least squares systems using the normal equations with ALIAS-C++, exploring the different operating modes of the library.
- Starting from a specific workbench code base written by the supervisor, implement a replacement for the current bound computation algorithm with one based on ALIAS-C++.
- Evaluate said implementation in terms of performance and the quality of the bounds for ranges of different size. Search performance can be compared against the existing bound, exhaustive searches and the PruneDIRECT algorithm. Study effects of adding additional constraints on the system (i.e. known constraints on the sum of coefficients in each row of the matrix), which is supported in ALIAS.
- Implement a minimum/maximum query quad tree for matrix elements in the existing R/qtl package (the relevant parts are written in C). Use the resulting intervals to implement a range search version of the R/qtl function `scantwo` for pairwise scans.

The first two items are the most crucial. They can also be implemented independent of the actual quality of ensuing results. If the project group would like to, some members could focus on tuning the problem formulation for a stricter bound and more efficient search process, while others focused on code aspects.

ALIAS is a C++ library. R/qtl is written in mostly C, but is an R package, so experience using R might also be beneficial for testing. The internal codebase is written in C++ with quite intensive use of Boost and the STL (including current use of the Boost Interval library). Familiarity with some of these technologies will make it easier to focus on the task itself, rather than the technical environment.

There is also an ALIAS-Maple version, which could be used in theoretical prototyping. However, the project supervisor cannot provide any Maple licenses, and has no experience in using ALIAS-Maple.

References

[1] Kajsa Ljungberg, Sverker Holmgren, and Örjan Carlborg. Simultaneous search for multiple qtl using the global optimization algorithm direct. *Bioinformatics*, 20(12):1887–1895, 2004.

[2] Carl Nettelblad, Behrang Mahjani, and Sverker Holmgren. Fast and accurate detection of multiple quantitative trait loci. *Journal of Computational Biology*, 20(9):687–702, 2013.

(C. Nettelblad) Laboratory of Molecular Biophysics, Department of Cell and Molecular Biology, Uppsala University, Sweden.

E-mail address: carl.nettelblad@icm.uu.se

[1] http://www-sop.inria.fr/coprin/logiciels/ALIAS/ALIAS-C++/ALIAS-C++.html