Analysis of optimization and numerical approaches to solve the linear least square problem

Emanuele Cosenza*, Riccardo Massidda†

Department of Computer Science
University of Pisa

* e.cosenza3@studenti.unipi.it, † r.massidda@studenti.unipi.it

***Abstract*—The linear least square problem can be tackled using a wide range of optimization or numerical methods. The L-BFGS method of the class of limited-memory quasi-Newton algorithms has been chosen for the former, whilst the thin QR factorization with Householder reflectors for the latter. Both these algorithms have been implemented from scratch using Python language, to finally experiment over their performances in terms of precision, stability and speed. The accordance of the implementations with the underlying theoretical models is also studied and discussed.**

Introduction

Given a dataset composed by a matrix $\hat{X} \in \mathbb{R}^{m \times n}$ with $m \geq n$ and a vector $y \in \mathbb{R}^m$, the solution of the linear least square (LLS) problem is the vector $w \in \mathbb{R}^n$ that fits best the data assuming a linear function between $\hat{X}$ and y. (Nocedal and Wright 2006, 50) This can be formalized as the following minimization problem:

$$w_* = \min_w \|\hat{X}w - y\|_2$$

The matrix $\hat{X}$ is actually composed in the following way:

$$\hat{X} = \begin{bmatrix} X^T \\ I \end{bmatrix}$$

Where $X \in \mathbb{R}^{n \times k}$ is a tall thin matrix, thus $m = k + n$. The LLS problem can be dealt both with iterative methods or with direct numerical methods. One algorithm has been chosen for each of these fields to finally discuss their experimental results.

L-BFGS

The L-BFGS is an iterative method of the quasi-Newton limited-memory class. It updates itself using the rule:

$$x_{k+1} = x_k - \alpha_k H_k \nabla f_k$$

The fact ??? differentiates it with the standard BFGS method. The initialization of the H_0^k matrix is fundamental and it is used to prove convergence results. How are the matrices updated. What line search do we use, how it is implemented.

Thin QR factorization

For the numerical counterpart, the thin QR factorization with Householder reflectors has been implemented as described in (Trefethen and Bau 1997).

By using the Householder QR factorization, the matrix R is constructed in place of $\hat{X}$ and the n reflection vectors $v_1, \ldots, v_n$ are stored. The reduced matrix $\hat{R}$ is trivially obtainable by slicing as in $\hat{R} = R_{1:n,1:n}$. In fact, given that $\hat{X}$ is already stored in memory and fully needed, there would be no advantage in directly constructing the reduced matrix.

By using the Householder vectors it is also possible to implicitly compute $\hat{Q}^T b$ to finally obtain w_* by back substitution over the upper-triangular system $\hat{R}w = \hat{Q}^T b$.

Algorithmic analysis

Convergence of L-BFGS

Liu and Nocedal (1989) defines three assumptions that lead to a theorem proving that the L-BFGS algorithm globally converges and moreover that it converges $\mathbb{R}$-linearly.

The first assumption is on the objective function f, that should be twice continuously differentiable. This is in fact true and we can define the gradient and the Hessian of the objective function as in:

$$\nabla f(w) = \hat{X}^T(\hat{X}w - y)$$

$$\nabla^2 f(w) = \hat{X}^T\hat{X}$$

Moreover the Hessian can be defined as a positive definite since it can be rearranged in the following way:

$$\begin{aligned} \nabla^2 f(w) &= \hat{X}^T\hat{X} \\ &= \begin{vmatrix} XI \end{vmatrix} \begin{vmatrix} X^T \\ I \end{vmatrix} \\ &= XX^T + I \end{aligned}$$

This also implies that the objective function f is a convex function and comes in handy for the second assumption

according to the sublevel set $D = \{w \in \mathbb{R}^n | f(w) \leq f(w_0)\}$ must be convex. Given that if a function is convex all of its sublevel sets are convex sets this is satisfied.

$$\begin{aligned}\forall x, y \in D \cap \lambda \in [0,1] \\ &\text{f convex} \\ &\implies f(\lambda x + (1-\lambda)y) \\ &\leq \lambda f(x) + (1-\lambda) f(y) \\ &\leq \lambda f(w_0) + (1-\lambda) f(w_0) \\ &= f(w_0) \\ &\implies \lambda x + (1 - \lambda y) \in D\end{aligned}$$

The third assumption requires that there exist two positive constant M_1 and M_2 such that $\forall z \in \mathbb{R}^n, w \in D$:

$$M_1 \|z\|^2 \leq z^T \nabla^2 f(w) \leq M_2 \|z\|^2$$

or equivalently

$$M_1 I \preceq \nabla^2 f(w) \preceq M_2 I$$

The first part of this equation is satisfied by the fact that the Hessian is positive definite. The second requires instead to satisfy

$$\begin{aligned}\nabla^2 f(w) \preceq M_2 I &\equiv X X^T + I \preceq M_2 I \\ &\equiv X X^T \preceq (M_2 - 1) I\end{aligned}$$

While this depends on the input data it should be noticed that the resulting matrix XX^T is symmetric, and therefore has n real eigenvalues, therefore it exists $\lambda_{max} = M_2 - 1$ to satisfy this third assertion.

Other then this assumptions, the theorem requires for the sequence of Hessian substitutes to be bounded. This clearly depends on the matrix chosen to initialize each run. What else can we say?

Armijo-Wolfe Line Search

The algorithm described in Al-Baali and Fletcher (1986) to perform an inexact line search is ensured to converge under the assumption that $\sigma > \rho$ (use the other notation here) since both the schemes are ensured to terminate.

Input data

Bibliography

Al-Baali, M., and R. Fletcher. 1986. "An Efficient Line Search for Nonlinear Least Squares." *Journal of Optimization Theory and Applications* 48 (3): 359–77. https://doi.org/10.1007/BF00940566.

Liu, Dong C., and Jorge Nocedal. 1989. "On the Limited Memory BFGS Method for Large Scale Optimization." *Mathematical Programming* 45 (1-3): 503–28. https://doi.org/10.1007/BF01589116.

Nocedal, Jorge, and Stephen J. Wright. 2006. *Numerical Optimization*. 2nd ed. Springer Series in Operations Research. New York: Springer.

Trefethen, Lloyd N., and David Bau. 1997. *Numerical Linear Algebra*. Philadelphia: Society for Industrial; Applied Mathematics.