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:

$$w_* = \min_w \|\hat{X}w - y\|_2^2$$

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. This method is actually a variation of the BFGS method, with which it shares the update rule; at the $i+1$-th iteration the point is updated as follows:

$$w_{i+1} = w_i - \alpha_i H_i \nabla f_i$$

L-BFGS has an inferior space complexity due to the fact that the Hessian approximation H_i is stored implicitly, and built over a fixed number of vector pairs $\{s_j, y_j\}$ of the previous t iterations and an initial matrix H_i^0. Where

$$s_i = w_{i+1} - w_i, \quad y_i = \nabla f_{i+1} - \nabla f_i$$
$$V_i = I - \rho_i y_i s_i^T, \quad \rho_i = \frac{1}{y_k^T s_k}$$

so H_i satisfies the following:

$$\begin{aligned} H_i = {} & (V_{i-1}^T \dots V_{i-t}^T) H_i^0 (V_{i-t} \dots V_{i-1}) \\ & + \rho_{i-t}(V_{i-1}^T \dots V_{i-t}^T + 1) s_{i-t} s_{i-m}^T (V_{i-t+1} \dots V_{i-1}) \\ & + \rho_{i-t+1}(V_{i-1}^T \dots V_{i-t}^T + 2) s_{i-t+1} s_{i-t+1}^T (V_{i-t+2} \dots V_{i-1}) \\ & + \dots \\ & + \rho_{i-1} s_{i-1} s_{i-1}^T \end{aligned}$$

Different strategies to initialize the H_i^0 matrix are proposed in the literature, and so they will be tested experimentally. Finally, the step size α_i is found by performing an inexact line search based on the Armijo-Wolfe conditions.

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 $v_1, \dots, v_n$ 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) define three necessary assumptions to prove a theorem stating that the L-BFGS algorithm globally converges and moreover that there exists a constant $0 \leq r < 1$ such that

$$f(w_i) - f(w_*) \leq r^i(f(w_0) - f(w_*))$$

so that the sequence w_i converges R-linearly.

The first assumption required 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 is positive definite, as can be easily seen by rearranging it in the following way:

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

Being the Hessian positive definite, the objective function f is a convex function. This comes in handy for the second assumption requiring the sublevel set $D = \{w \in \mathbb{R}^n | f(w) \leq f(w_0)\}$ to be convex, it can be easily proved that if a function is convex all of its sublevel sets are convex sets.

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

The third and last assumption requires the existence of two positive constants M_1 and M_2 such that $\forall z \in \mathbb{R}^n, w \in D$:

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

or equivalently

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

Since $\nabla^2 f(w)$ is positive definite the previous condition is true for $M_1 = \lambda_{min}$ and $M_2 = \lambda_{max}$.

Other then these assumptions, the theorem requires for the sequence of Hessian substitutes $\{H_i\}$ to be bounded. This obviously depends on the initialization technique used to generate H_i^0, various techniques are suggested in the literature such as $H_k^0 = \gamma_k I$ or $H_k^0 = \gamma_k H_0$ where

$$\gamma_k = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|}$$

Other initialization techniques may possibly be tested and evaluated experimentally.

In the convergence proof the M_2 constant is used to upper bound the trace of the next Hessian substitute H_{i+1} by using a derived constant M_3.

$$tr(H_{i+1}) \leq tr(H_i^0) + tM_2 \leq M_3$$

implying that the largest eigenvalue of H_{i+1} is always constrained under a fixed constant. On the other hand the M_1 constant is used, always by using a derived constant M_4, to lower bound the determinant of H_{i+1}.

$$det(H_{i+1}) \geq det(H_i^0) + (\frac{M_1}{M_3})^t \geq M_4$$

Implying in this case that the smallest eigenvalue of H_{i+1} is always positive. These reasoning is used to prove this fundamental assertion

$$\exists \delta > 0 : \cos\theta_i = \frac{s_i^T H_i s_i}{\|s_i\|\|H_i s_i\|} \geq \delta$$

That with the other assumptions leads to the R-linearly convergence result previously stated. Moreover if the constant M_1 was to be equal to zero, given that $\lambda_{min} = 0$, then also M_4 constant may be equal to zero, and this is not enough to prove the existence of $\delta > 0$ for each step. Anyhow, given the structure of $\nabla^2 f(x)$ we can prove that $\lambda_{min} > 0$. The matrix XX^T is positive semi-definite, since $\forall z : z^T XX^T z = \|X^T z\| \geq 0$, therefore all the eigenvalues of the matrix are non-negative. Furthermore, according to the spectral theorem, since XX^T is symmetric, there exists U orthogonal matrix and D diagonal containing the eigenvalues of XX^T.

$$\begin{aligned}\nabla^2 f(x) &= XX^T + I \\ &= UDU^T + I \\ &= UDU^T + UIU^T \\ &= U(D+I)U^T\end{aligned}$$

Therefore $\lambda_{min} \geq 1$.

Armijo-Wolfe inexact line search

The convergence proof requires the algorithm to perform a line search respectful of the Armijo-Wolfe conditions, the solution described in Al-Baali and Fletcher (1986) is therefore adapted and implemented.

The algorithm performs an inexact line search that is ensured to converge under the assumption that $\sigma > \rho$ where $\rho \in (0, \frac{1}{2}), \sigma \in (0, 1)$, respectively the constant for the Armijo condition and for the Wolfe one. By defining the function ϕ, used to evaluate the value of f at a certain step-size α, the conditions can be defined as follows.

$$\begin{aligned} \phi(\alpha) &= f(w_i + \alpha d_i) \\ \phi(\alpha) &\leq \phi(0) + \alpha\rho\phi'(0) \qquad \text{(A)} \\ \phi'(\alpha) &\geq \sigma\phi'(0) \qquad \text{(W)} \end{aligned}$$

The algorithm requires a lower bound $\bar{f}$ on $\phi(\alpha)$ for $\alpha \geq 0$. More precisely, it assumes that the user is prepared to accept any value of α for which $\phi(\alpha) \leq \bar{f}$ where $\bar{f} < \phi(0)$. For the linear least-squares problem an obvious lower bound is $\bar{f} = 0$.

The algorithm performs then an inexact line search by searching a candidate point α_i at the i-th iteration in the interval (a_i, b_i), stopping if such candidate reaches the lower bound or if it satisfies both (A) and (W).

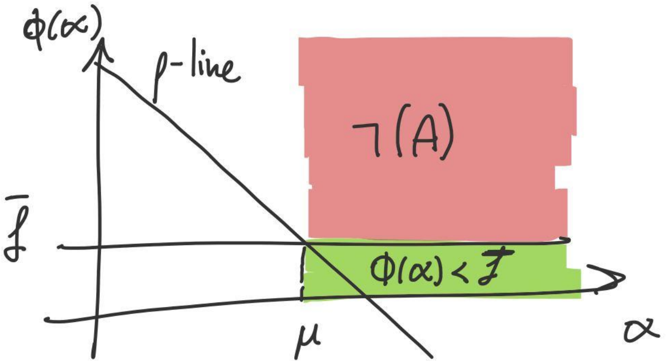

Fig. 1. Graphical depiction of the μ point.

The Armijo condtion describes a line, called ρ-line, in the plot $(\alpha, \phi(\alpha)$ that can be useful to bound the starting interval. In fact the initial search interval can be reduced from $(0, \infty)$ to $(0, \mu)$ where

$$\mu = \frac{\bar{f} - \phi(0)}{\rho\phi'(0)}$$

It is immediate that $\forall\alpha > \mu$ either (A) can't be satisfied, or the point lies under the lower bound $\bar{f}$ (figure 1).

To proceed with the discussion over the shrinking procedure the function T is defined as in

$$T(a, b) = [a + \tau_1(b - a), b - \tau_2(b - a)]$$

where $0 < \tau_1 \leq \tau_2 \leq \frac{1}{2}$.

If the candidate doesn't satisfy (A) or if the left extreme a_i constitues a better point, the next candidate is chosen in the interval $T(a_i, \alpha_i)$. Otherwise if the candidate doesn't satisfies (W) the next candidate is chosen in $T(\alpha_i, b_i)$. In both cases the (a_{i+1}, b_{i+1}) are updated with the extremes returned by the T function.

The candidate step-size may be randomly chosen between all the points in the interval defined by the T function, this approach will be experimentally tested against quadratic interpolation.

It should be noted that the Al-Baali and Fletcher (1986) paper defines the function T in a slightly different way, together with another function E used to specifically define the interval when (W) is not satisfied. The simplification hereby described is due to the fact that in our implementation it is ensured that $\forall i : a_i \leq \alpha_i \leq b_i \wedge b_i \neq \infty$, moreover this does not interfere with the convergence proof.

Input data

Implementation details

Experimental results

Conclusions

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.