CGN: A Gauss-Newton method for solving penalized nonlinear least-squares problems with linear constraints.

F. Parzer

June 24, 2021

Abstract

1 Introduction

We implement a Gauss-Newton method that solves equality- and inequality constrained nonlinear least-squares problems with multiple regularization terms. That is, the method is applicable to all problems which can be written as

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & \frac{1}{2} \left\| F(x_1, \ldots, x_l) \right\|_2^2 + \frac{1}{2} \left(\sum_{i=1}^{l} \beta_i \left\| P_i (x_i - \bar{x}_i) \right\|_2^2 \right), \\
\text{s. t.} \quad & Ax = b, \\
& Cx \geq d, \qquad\qquad x_i \in \mathbb{R}^{n_i}, \quad n = \sum_i n_i.
\end{aligned} \tag{CNLS}
$$

where we call the term $\|F(x)\|_2^2$ the **misfit term**, and the terms $\|P_i(x_i - \bar{x}_i)\|_{p_i}^{p_i}$ the **regularization terms**, and

- $F : \mathbb{R}^n \to \mathbb{R}^{m_1}$ is a continuously differentiable **misfit function**. As an example, in nonlinear regression we typically have

  $$F(x) = \Gamma^{-1/2}(\mathcal{G}(x) - y),$$

  where $\mathcal{G}$ is a nonlinear observation operator,y a given measurement, and $\Gamma \in \mathbb{R}^{m_1 \times m_1}$ the covariance matrix of the measurement noise;

- $\beta > 0$ is a tunable regularization parameter that determines how strongly the regularization terms are weighted with respect to the data term. There are iterative strategies of choosing β in case the user does not want to choose it manually;

- $P_i \in \mathbb{R}^{r_i \times n_i}$ are weighting matrices of rank n_i. We assume throughout that $r_i \geq n_i$ and that P_i has full column rank, which implies that a left-inverse P_i^+ of P_i is avaiable, i.e. that

  $$P_i^+ P_i = \mathbb{I}_{n_i}.$$

  In the case of MAP estimation with Gaussian priors ($p_i = 2$), $(P_i P_i^\top)^{-1}$ corresponds to the **prior covariance**. We allow the case where $r_i \neq n_i$, since it is relevant for compressed-sensing applications, where P_i will map into a large redundant dictionary, and we have $r_i \gg n_i$.

- $\bar{x}_i \in \mathbb{R}^{n_i}$ is an initial guess which regularizes our fit. In the case of MAP estimation, it corresponds to the **prior mean**;

- $A \in \mathbb{R}^{c_1 \times n}$ and $b \in \mathbb{R}^{c_1}$ define equality constraints.

- $C \in \mathbb{R}^{c_2 \times n}$ and $d \in \mathbb{R}^{c_2}$ define the inequality constraints.

- We will assume throughout that the problem is feasible, i.e. the set $\{ x \in \mathbb{R}^n : Ax = b, Cx \geq d \}$ is nonempty.

The `CGN` package is a library for solving problem (CNLS) using a generalized version of the Gauss-Newton method.

2 Main loop

2.1 Getting rid of the equality constraints

First, we explain how the assumption that A has full rank means that we can assume wlog that no equality constraints are present.

Since A is by assumption of full rank, we can compute its QR-decomposition

$$\begin{aligned} A^\top &= \begin{bmatrix} Q & S \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}, \\ &\text{where } Q \in \mathbb{R}^{n \times c_1}, \quad S \in \mathbb{R}^{n \times (n-c_1)}, \end{aligned}$$

and where $[Q, S] \in \mathbb{R}^{n \times n}$ is an orthogonal matrix and $R \in \mathbb{R}^{c_1 \times c_1}$ is an invertible upper triangular matrix. Then, we define a new variable $q \in \mathbb{R}^n$ by

$$\begin{bmatrix} q \\ s \end{bmatrix} = [Q, S]^\top x = \begin{bmatrix} Q^\top x \\ S^\top x \end{bmatrix}, \quad q \in \mathbb{R}^{c_1}, s \in \mathbb{R}^{n-c_1}.$$

Then, the constraint $Ax = b$ becomes

$$\begin{aligned} & \begin{bmatrix} R^\top & 0 \end{bmatrix} \begin{bmatrix} Q^\top \\ S^\top \end{bmatrix} x = b, \\ \Leftrightarrow \quad & R^\top Q^\top = b \\ \Leftrightarrow \quad & R^\top q = b. \end{aligned}$$

Consequently q^* is determined by $R^\top q^* = b$.

We then reduce the minimization with respect to x to the free variable s through the identity

$$x = [Q, S][Q, S]^\top x = [Q, S] \begin{bmatrix} q \\ s \end{bmatrix} = Qq + Ss.$$

The fully constrained least-squares problem (CNLS) is consequently equivalent to the inequality-constrained problem

$$\begin{aligned} \min_{s \in \mathbb{R}^{n-c_1}} \quad & \frac{1}{2} \left\| \tilde{F}(s) \right\|^2 + \frac{\beta}{2} \left\| \tilde{P}s - \tilde{s} \right\|^2, \\ & \tilde{C}s \geq \tilde{d}, \\ \text{where} \quad & \tilde{F}(s) = F(x_1^* + Ss), \quad \tilde{P} = PS, \quad \tilde{s} = \bar{x} - x_1^*, \\ & \tilde{C} = CS, \quad \tilde{d} = d - x_1^*. \end{aligned} \tag{2.1}$$

If s^* is a minimizer of the reduced problem, then

$$x^* = x_1^* + Ss^*$$

is a minimizer for (CNLS).

2.2 Main loop

By the preceding section, we can consider without loss of generality the reduced problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \frac{1}{2}\left\|F(x)\right\|_2^2 + \frac{1}{2}\left\|Px - \tilde{x}\right\|_2^2, \\ \text{s. t.} \quad & Cx \geq d. \end{aligned} \tag{CNLS2}$$

The main idea to solve this problem is consecutive linearization. Given the current iterate x^k, we define

$$\begin{aligned} F_k &:= F(x^k), \\ J_k &:= F'(x^k), \end{aligned}$$

and solve the linearized problem

$$\begin{aligned} \Delta x^{k+1} = \operatorname*{argmin}_{\Delta x\in\mathbb{R}^n} \quad & \frac{1}{2}\left\|F_k + J_k\Delta x\right\|_2^2 + \frac{1}{2}\left\|P\Delta x + Px^k - \tilde{x}\right\|_2^2, \\ \text{s. t.} \quad & C\Delta x \geq d - Cx^k. \end{aligned} \tag{CLS}$$

Then we define a search path

$$x^{k+1}(h) := x^k + h\Delta x^{k+1}, \qquad h > 0$$

and choose a step size h^k using a line search. The next iterate is then

$$x^{k+1} := x^{k+1}(h^k),$$

and the procedure is repeated.

2.3 Solving (CLS)

Most of this section is based on the classical reference on least-squares problems by Lawson and Hanson [2].

It remains to solve the inequality-constrained least-squares problem (CLS). It is easy to check that (CLS) can be brought into the form

$$\begin{aligned} \min_{x} \quad & \frac{1}{2}\left\|Gx - h\right\|_2^2 \\ \text{s. t.} \quad & Cx \geq f, \end{aligned} \tag{LSI}$$

by setting

$$G = \begin{bmatrix} J_k \\ P \end{bmatrix}, \quad h = \begin{bmatrix} -F_k \\ \tilde{x} - Px^k \end{bmatrix}, \quad f = d - Cx^k.$$

This problem is an **inequality-constrained linear least-squares problem (LSI)**, for which a solution can be obtained by solving an associated **nonnegative least-squares problem**. We describe next how this is done.

2.3.1 From LSI to LDP

Since P has full column rank by assumption, so does G. Hence, G admits a QR decomposition of the form

$$G = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} T \\ 0 \end{bmatrix},$$

where T is a square upper triangular matrix of full rank. If we plug this decomposition in the objective function of problem (LSI), we have

$$\begin{aligned} \left\|Gx - h\right\|_2^2 &= \left\|U^\top(Gx - h)\right\|_2^2 \\ &= \left\|Tx - U_1^\top h\right\|_2^2 + \left\|U_2^\top h\right\|_2^2. \end{aligned}$$

The second term is constant and can therefore be ignored in the optimization. Making the change of variables

$$u = Tx - U_1^\top h \quad \iff \quad x = T^{-1}(u + U_1^\top h).$$

problem (LSI) is now equivalent to

$$\begin{aligned} \min_u \quad & \frac{1}{2}\|u\|_2^2 \\ \text{s. t.} \quad & Ku \geq l \\ \text{where} \quad & K = CT^{-1}, \quad l = d - KU_1^\top h. \end{aligned} \tag{LDP}$$

This is a **least distance problem (LDP)**, for which a solution can be obtained by solving an associated nonnegative least-squares problem (see next section). The solution x^* of problem (LSI) can then be computed by reverting the change of variables. That is, if u^* solves (LDP), then

$$x^* = T^{-1}(u^* + U_1^\top h)$$

solves (LSI).

2.3.2 From LDP to NNLS

Finally, it can be shown (see [2, chapter 23.4]) that the LDP problem

$$\begin{aligned} \min_u \quad & \frac{1}{2}\|u\|_2^2 \\ & Ku \geq l \end{aligned}$$

is equivalent to the following **nonnegative least-squares problem (NNLS)**:

$$\begin{aligned} \min_v \quad & \frac{1}{2}\|Mv - e_{n+1}\|_2^2 \\ \text{s.t.} \quad & v \geq 0, \\ & \text{where } M = \begin{bmatrix} K^\top \\ l^\top \end{bmatrix}, \text{ and } e_{n+1} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \in \mathbb{R}^{n+1}. \end{aligned} \tag{NNLS}$$

If v_* solves this problem, then the solution u_* of (LDP) is given by

$$u_i^* = -\frac{r_i^*}{r_{n+1}^*}, \quad i = 1, \ldots, n,$$

where $r^* = Mv^* - e_{n+1}$ is the residual of problem (NNLS).

An algorithm for solving (NNLS) is provided in [**LawHan87**]. At the moment, we use the implementation of the Lawson-Hanson algorithm given by `scipy.optimize.nnls`. In the future, we might substitute this with the faster **FNNLS** algorithm by Bro and de Jong [1]. However, since the time of computing the QR decompositions usually outweighs the solution of the NNLS problem, this is not a priority.

2.4 Line search with the Wächter-Biegler filter

Due to the presence of constraints, we cannot use the same line search strategy for the generalized Gauss-Newton method as in the unconstrained case. Instead, we will use the line search filter by Wächter and Biegler. We provide a rough outline and pseudocode of the method, and refer to the original paper [**WaeBie03**] for further details.

The idea of a line search filter is to start with a stepsize $h_0 > 0$ and then decrease the steplenght until there is either sufficient decrease in the objective function ϕ or in an infeasibility measure, which quantifies how well the proposed iterate satisfies the constraints.

For our case of mixed equality and inequality constraints, we define the infeasibility measure

$$\theta(x) = \|Ax - b\|_1 + |Cx - d|^- ,$$

where

$$|z|^- = \sum_{i=1}^{n} \max(0, -z_i), \qquad \text{for } z \in \mathbb{R}^n.$$

A new step $w_{k+1}(h) = w_k + h\Delta w_k$ is then accepted if

$$\begin{aligned} &\theta(w_{k+1}(h)) \leq (1-\gamma_\theta)\theta(w_k), \\ \text{or} \quad &\phi(w_{k+1}(h)) \leq \phi(w_k) - \gamma_\phi \theta(w_k), \end{aligned}$$

where ϕ is the least-squares cost function defined in **??**, while $\gamma_\theta \in (0, 1)$ and $\gamma_\phi \in (0, 1)$ are small constants which determine what constitutes "sufficient decrease".

Furthermore, there is a switching condition which ensures that the algorithm still reduces the objective function. A stepsize h meets the switching condition if

$$hm_k < 0 \quad \text{and} \quad (-hm_k)^{s_\varphi} h^{1-s_\varphi} > \delta\theta(w_k)^{s_\theta}, \tag{2.2}$$

where $m_k = \phi'(w_k)\Delta w_k$ and $\delta > 0$, $s_\theta > 1$ and $s_\varphi > 2s_\theta$ are tunable parameters. If the switching condition is satisfied, we enforce sufficient decrease in the objective function by accepting the step if and only if

$$\phi(w_{k+1}(h)) \leq \phi(w_k) + \eta m_k(h), \tag{2.3}$$

where $\eta \in (0, \frac{1}{2})$ is a small constant.

Finally, we have to keep track of a name-giving filter, which ensures that there are no cycles in the trajectory of our optimization method. A new step is only accepted if it satisfies

$$\begin{aligned} &\theta(w_{k+1}(h)) < (1-\gamma_\theta)\theta(w), \\ \text{or} \quad &\phi(w_{k+1}(h)) < \phi(w) - \gamma_\phi \theta(w), \end{aligned}$$

for all iterates $w \in \mathcal{F}$ in the filter. The new iterate w_{k+1} is then added to the filter if it did not satisfy the switching condition or the Armijo condition.

If no acceptable steplength can be found, a feasibility restoration phase is invoked which computes a new iterate w_{k+1} that is close to the last iterate w_k but also satisfies all constraints. This happens if the stepsize h is decreased below the value

$$h_k^{\min} := \gamma_h \cdot \begin{cases} \min\{\gamma_\theta, \frac{\gamma_\phi \theta(w_k)}{-m_k}, \frac{\delta\theta(w_k)^{s_\theta}}{(-m_k)^{s_\phi}}\}, & \text{if } m_k < 0, \\ \gamma_\theta, & \text{otherwise.} \end{cases}$$

Algorithm 1 `linesearchFilter`

Given w_k, Δw_k, the current filter $\mathcal{F}$, and tunable constants $c \in (0,1)$, $\gamma_\theta, \gamma_\phi \in (0,1)$, $\delta > 0$, $s_\theta > 1$, $s_\varphi > 2s_\theta$, $\eta_\phi \in (0, \frac{1}{2})$ and $\gamma_h \in (0,1)$.

$h = 1$;
$m_k = \phi'(w_k)\Delta w_k$;
$h^{\min} = \gamma_h \cdot \begin{cases} \min\{\gamma_\theta, \frac{\gamma_\phi \theta(w_k)}{-m_k}, \frac{\delta \theta(w_k)^{s_\theta}}{(-m_k)^{s_\phi}}\}, & \text{if } m_k < 0, \\ \gamma_\theta, & \text{otherwise.} \end{cases}$
repeat
  $w_{k+1} = w_k + h\Delta w_k$;
  if $hm_k < 0$ and $(-hm_k)^{s_\varphi} h^{1-s_\varphi} > \delta \theta(w_k)^{s_\theta}$ **then**
    if $\phi(w_{k+1}) \leq \phi(w_k) + \eta_\phi h m_k$ **then**
      if $\theta(w_{k+1}) \leq (1-\gamma_\theta)\theta(w)$ or $\phi(w_{k+1}) \leq \phi(w) - \gamma_\phi \theta(w)$ for all $w \in \mathcal{F}$ **then**
        break;
      end if
    end if
  else if $\theta(w_{k+1}) \leq (1-\gamma_\theta)\theta(w_k)$ or $\phi(w_{k+1}) \leq \phi(w_k) - \gamma_\phi \theta(w_k)$ **then**
    if $\theta(w_{k+1}) \leq (1-\gamma_\theta)\theta(w)$ or $\phi(w_{k+1}) \leq \phi(w) - \gamma_\phi \theta(w)$ for all $w \in \mathcal{F}$ **then**
      if $\phi(w_{k+1}) > \phi(w_k) + \eta_\phi h m_k$ **then**
        add w_{k+1} to $\mathcal{F}$;
      end if
      break;
    end if
  end if
  $h = c \cdot h$;
until $h \leq h^{\min}$
if $h \leq h^{\min}$ **then**
  find w_{k+1} using feasibility restoration;
  add w_k to $\mathcal{F}$;
end if
return w_{k+1} and $\mathcal{F}$;

References

[1] R. Bro and S. De Jong. "A fast non-negativity-constrained least squares algorithm". In: *Journal of Chemometrics* 11.5 (1997), pp. 393–401. ISSN: 0886-9383. DOI: 10.1002/(sici)1099-128x(199709/10)11:5<393::aid-cem483>3.0.co;2-l. URL: https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1099-128X(199709/10)11:5%5C%3C393::AID-CEM483%5C%3E3.0.CO;2-L (cited on page 4).

[2] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall Series in Automatic Computation. Prentice-Hall, 1972 (cited on pages 3, 4).