Notes on Nocedal and Wright's "Numerical Optimization"
Chapter 10 – "Least Squares Methods"

Lucas N. Ribeiro

Introduction I

- Least squares problems:

$$f(x) = 0.5 \sum_{j=1}^{m} r_j^2(x)$$

  where r_j is the residual function from $\mathbb{R}^N$ to $\mathbb{R}$.

- Its gradient (notice that $f(x)$ is a summation of *quadratic* functions):

$$\nabla f(x) = \sum_{j=1}^{m} r_j(x) \nabla r_j(x) = J(x)^{\mathsf{T}} r(x)$$

Introduction II

- Its Hessian (apply the product rule to $\nabla f(x)$):

$$\nabla f^2(x) = \sum_{j=1}^{m} \nabla r_j(x) \nabla r_j(x)^\mathsf{T} + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x) \tag{1}$$

$$= J(x)^\mathsf{T} J(x) + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x) \tag{2}$$

- The Jacobian matrix: $J(x) = [\partial r_j / \partial x_i]$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.
- Once we have the Jacobian, we can easily calculate the gradient and the Hessian.
- The second term in the Hessian is usually unimportant (because the residuals are small – *if the model fits!*)
- Solutions to minimize $f(x)$ are usually based on line search and trust region methods which exploit the available structure

Linear Least Squares I

- Model: $\phi(x; t_j)$
- Measurement: y_j
- Residual: $r_j(x) = \phi(x; t_j) - y_j$
- In the linear least squares problem, the model is linear and the residual can be written as $r(x) = Ax - y$, for some A matrix.
- The objective function is now $f(x) = 0.5\|Ax - y\|^2$
- Its gradient: $\nabla f(x) = A^\mathsf{T} Ax - Ay$
- Its Hessian: $\nabla^2 f(x) = A^\mathsf{T} A$ (note that the second term vanished)
- By setting $\nabla f(x) = 0$, we have the normal equations:

$$A^\mathsf{T} Ax^* = J^\mathsf{T} y$$

- One can solve the normal equation by Cholesky, QR or SVD factorization, in which the latter is the most robust

Linear Least Squares II

- The SVD (of A) solution is given by (pseudo-inverse)

$$x^* = \sum_{i=1}^{n} \frac{u_i^\mathsf{T} y}{\sigma_i} v_i$$

- When the problem is too large, CG may be useful to solve the normal equations

Non-Linear Least Squares – Gauss-Netwon

- Modified Newton's method with line search
- Remember, Newton's method solves: $\nabla^2 f(x_k)p_k = -\nabla f(x_k)$.
- We consider the approximation:

$$\nabla^2 f(x_k) \approx J_k^\mathsf{T} J_k \tag{3}$$

- And the GN method solves

$$J_k^\mathsf{T} J_k p_k^{GN} = -J_k^\mathsf{T} r_k$$

- It's similar to the normal equation $\rightarrow$ recast as linear LS problem!
- For each step, solve:

$$\min_p 0.5\|J_k p + r_k\|^2$$

- Can be solved by means of SVD, QR, CG method, etc
- Nice performance if J_k has full rank.

Non-Linear Least Squares – Levenberg-Marquardt

- Newton's method approximation with trust-region method
- Works better than GN when the Jacobian is rank-deficient
- Formulation:

$$\min_p 0.5\|J_k p + r_k\|^2, \quad \text{subject to } \|p\| \leq \Delta_k \tag{4}$$

- When the GN solution lies in the trust region, it's the solution ($\|p^{GN}\| < \Delta$)
- Otherwise, there is a $\lambda > 0$ such that the solution of the LS problem $(J^\mathsf{T} J + \lambda I)p = -J^\mathsf{T} r$ (this problem is obtained by incorporating the trust region into the objective function by a Lagrange multiplier)
- A λ that matches Δ can be found by a root-finding algorithm

Non-Linear Least Squares – Orthogonal Distance Regression

- Incorporate errors within the model

$$y_j = \phi(x; t_j + \delta_j) + \epsilon_j$$

- define the minimization problem

$$\min_{x,\delta_j,\epsilon_j} 0.5 \sum_{j=1}^{m} w_j^2 \epsilon_j^2 + d_j^2 \sigma_j^2$$

- If all the weights are equal: shortest distance $\rightarrow$ (it's orthogonal to the curve at the point of intersection)
- can be reformulated as a linear least squares problem