Reference
Numerical Optimization by J. Nocedal and S. Wright, 2^{nd} Ed., *Chapt.*10

Least-Squares Problems

Problem Formulation

One of the special forms of least-square problems which is easier to solve has the following objective function:

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

Notations and definitions:

- j^{th} input instance is given by t_j
- The original function is given by $y(t_j)$
- The hypothesis (model) is given as $\phi(x; t_j)$
- x is a parameter vector of hypothesis
- $x \in \mathcal{R}^n$ and for least-squares $m >= n$
- The discrepancy (alternative names are residual and error) is given as

$$r_j(x) = \phi(x; t_j) - y(t_j) \tag{2}$$

Here, r_j is a smooth function such that $r_j : \mathcal{R}^n \to \mathcal{R}$. We aim to minimize objective function (1) to estimate the parameters of the best fit model.

A residual vector $r : \mathcal{R}^n \to \mathcal{R}^m$ is formed after stacking each residual r_j and given as follows:

$$r(x) = (r_1(x), r_2(x), \ldots, r_m(x))^T \tag{3}$$

It allows us to write (1) as $f(x) = \frac{1}{2} \parallel r(x) \parallel_2^2$

Gradient and Hessian of $f(x)$

- Gradient

$$\nabla f(x) = \sum_{j=1}^{m} r_j(x)\nabla r_j(x) = \nabla r(x)^T r(x) = J(x)^T r(x) \tag{4}$$

where, $\nabla r(x) = J(x)$ is $m \times n$ *Jacobian* matrix of residuals and given as:

$$J(x)_{m\times n} = \underbrace{\left[\frac{\partial r_j}{\partial x_i}\right]}_{j=1,2,\ldots,m\ i=1,2,\ldots,n} = \begin{bmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{bmatrix} \tag{5}$$

- Hessian

$$\nabla^2 f(x) = \sum_{j=1}^{m} \nabla r_j(x) \nabla r_j(x)^T + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x) = J(x)^T J(x) + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x) \tag{6}$$

  Structural properties of Hessian are as follows:

  - Jacobian matrix computation is easy and inexpensive
  - Second term in (6) is close to zero in many practical applications either because $\nabla^2 r_j(x)$ are relatively small near the solution or residuals $r_j(x)$ are relatively smaller, making first term more dominant.

Relationship with Likelihood Estimation

Let discrepancies are given by $\epsilon_j = \phi(x; t_j) - y_j$. Here, model $\phi(x; t_j)$ is linear function of x. Let us assume ϵ_j's are independent and identically distributed (*iid*) with variance σ^2 and probability density is given by $g_\sigma(\cdot)$. The likelihood of set of observations $y_j, j = 1, 2, \ldots, m$ is given as:

$$p(y; x, \sigma) = \prod_{j=1}^{m} g_\sigma(\epsilon_j) = \prod_{j=1}^{m} g_\sigma(\phi(x; t_j) - y_j) \tag{7}$$

When discrepancies are normally distributed,

$$g_\sigma(\epsilon) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-\epsilon^2}{2\sigma^2}\right) \tag{8}$$

With above value of $g_\sigma(\epsilon)$ (7) becomes

$$p(y; x, \sigma) = (2\pi\sigma^2)^{\frac{-m}{2}} \exp\left(\frac{-1}{2\sigma^2} \sum_{j=1}^{m} [\phi(x; t_j) - y_j]^2\right) \tag{9}$$

The log likelihood is given as,

$$\log(p(y;x,\sigma)) = \frac{-m}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{j=1}^{m}[\phi(x;t_j) - y_j]^2 \tag{10}$$

From (10), one can conclude that maximization of likelihood function under a conditional Gaussian noise distribution for a linear model is equivalent to minimizing sum-of-square error function given by $\frac{1}{2}\sum_{j=1}^{m}[\phi(x;t_j) - y_j]^2 = \frac{1}{2}\sum_{j=1}^{m}\epsilon_j^2$.

Linear Least-Square Problems

From (2), residual $r_j(x)$ is linear function of x, thus minimizing (1) is called a *linear least-square problem*. The residual vector can be written as $r(x) = Jx - y$ for some matrix J and vector y,then the objective function is

$$f(x) = \frac{1}{2} \parallel Jx - y \parallel^2 \tag{11}$$

Here, $y = r(0)$, $\nabla f(x) = J^T(Jx - y)$ and $\nabla^2 f(x) = J^T J$ (as r_j is linear). As (11) is convex, the global minimizer x^* for which $\nabla f(x^*) = 0$, should satisfy following *normal equations*

$$J^T Jx^* = J^T y = -J^T r \tag{12}$$

Assuming $m \geq n$ and J has full column rank, unconstrained linear-least square problem can be solved using following algorithms:

1. Cholesky decomposition 2. QR decomposition 3. SVD

Cholesky decomposition

As $J^T J$ is symmetric matrix, it is decomposed as $J^T J = R^T R$, where R is upper triangular matrix. Therefore, $J^T Jx = R^T Rx = R^T z = J^T y$, where $z = Rx$. The parameter x is determined by taking two back-substitutions. This algorithm is not useful when J is ill-conditioned as relative error is proportional to square of the condition number.

QR decomposition

The Euclidean norm of any vector is not affected by orthogonal transformations, so for any $m \times m$ orthogonal matrix Q, we can write

$$\parallel Jx - y \parallel = \parallel Q^T(Jx - y) \parallel \tag{13}$$

After performing QR factorization with pivoting on matrix J, we get

$$J\Pi = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R_1 \tag{14}$$

where,

- $\Pi_{n\times n}$ is a permutation matrix(hence guaranteed to be orthogonal)
- $Q_{m\times m}$ is an orthogonal matrix
- Q_1 contains the first n columns of Q
- Q_2 contains the last $m-n$ columns of Q
- $R_{n\times n}$ is the upper triangular matrix with positive diagonal elements

Using (13) and (14),

$$\begin{aligned} \parallel Jx - y \parallel_2^2 &= \left\| \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} (J\Pi\Pi^T x - y) \right\|_2^2 = \left\| \begin{bmatrix} R \\ 0 \end{bmatrix} (\Pi^T x) - \begin{bmatrix} Q_1^T y \\ Q_2^T y \end{bmatrix} \right\|_2^2 \\ &= \left\| R(\Pi^T x) - Q_1^T y \right\|_2^2 + \left\| Q_2^T y \right\|_2^2 \end{aligned} \tag{15}$$

Second term $\left\|Q_2^T y\right\|_2^2$ in (15) is independent of x, therefore, $\parallel Jx-y \parallel$ is minimized by driving first term in (15) to zero, giving $x^* = \Pi R^{-1} Q_1^T y$

This algorithm has relative error proportional to the condition number of J.

SVD

SVD of J is given as

$$J = U \begin{bmatrix} S \\ 0 \end{bmatrix} V^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S \\ 0 \end{bmatrix} V^T = U_1 S V^T \tag{16}$$

where,

- $U_{m\times m}$ is an orthogonal matrix
- U_1 contains the first n columns of U
- U_2 contains the last $m-n$ columns of U
- $V_{n\times n}$ is an orthogonal matrix
- $S_{n\times n}$ is a diagonal matrix, with diagonal elements $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_n \geq 0$

Therefore,

$$\parallel Jx - y \parallel_2^2 = \left\| \begin{bmatrix} S \\ 0 \end{bmatrix} (V^T x) - \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} y \right\|_2^2 = \parallel S(V^T x) - U_1^T y \parallel_2^2 + \parallel U_2^T y \parallel_2^2 \tag{17}$$

Now, driving first term in (17) to zero, optimum is $x^* = VS^{-1}U_1^T y = \sum_{i=1}^{n} \frac{u_i^T y}{\sigma_i} v_i$

Comparison of Cholesky decomposition, QR decomposition and SVD

Cholesky decomposition

- The condition number of J^TJ is the square of the condition number of J. This can lead to less accurate solution compared to methods that avoid squaring of condition number
- When J is ill conditioned, due to round-off errors, small negative values can appear on the diagonal during factorization process resulting into poor solutions
- Useful when $m >> n$. Less expensive when $m >> n$ and J is sparse

QR decomposition

- The condition number of the problem is equal to condition number of J and not degraded unlike Cholesky decomposition
- It provides limited information about the sensitivity of the solution to perturbations in the data (J or y)
- When compared to Cholesky decomposition, this algorithm is computationally expensive and more numerically robust

SVD

- Provides sensitivity information
- This algorithm is the most expensive in computations and the most robust and reliable of all

Non Linear Least-Square Problems

- The Gauss-Newton Method
- The Levenberg-Marquardt Method

The Gauss-Newton Method (GN)

This method can be viewed as modified Newton's method with line search. The standard Newton equation is $\nabla^2 f(x_k)p = -\nabla f(x_k)$. In GN method, instead of using standard Newton equation, the search direction p_k^{GN} is obtained as follows:

$$J_k^T J_k p_k^{GN} = -J_k^T r_k \tag{18}$$

Above formulation has following advantages:

- $\nabla^2 f_k(p) \approx J^TJ$ is computationally inexpensive

- In most of the practical cases, J^TJ is dominant over second term in (6), thus second term is ignored
- If J_k is full rank and $\nabla f_k \neq 0$ then p_k^{GN} is a descent direction as follows:

$$(p_k^{GN})^T \nabla f_k = (p_k^{GN})^T J_k^T r_k = -(p_k^{GN})^T J_k^T J_k p_k^{GN} = - \parallel J_k p_k^{GN} \parallel_2^2 \leq 0 \tag{19}$$

- By comparing (12) and (18), p_k^{GN} is solution of the following linear least-squares problem

$$\min_p \frac{1}{2} \parallel J_k p + r_k \parallel_2^2 \tag{20}$$

  Thus, search direction p_k^{GN} can be obtained using linear least-square algorithms.

Recall that the Gauss-Newton method is based on line search and the search direction p_k^{GN} is a valid descent direction only if J_k is full rank. This limitation is handled in the Levenberg-Marquardt method using trust region approach.