Least Squares and Its Variant

Pramook Khungurn

June 5, 2013

In this note, scalars are denoted by small letters in normal type face; for examples, x, y, and z. Vectors are denoted by small letters in bold face; for examples, $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$. Entries of vectors are denoted by the same small latter in normal type face as the letter representing the vector; for example, $\mathbf{x} = (x_1, x_2, x_3)$. Matrices are denoted by capital letters in bold face such as $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$, and we use the same small letter to the entries. For example, the entries of $\mathbf{X}$ are x_{11}, x_{12}, and so on.

After reading books and internet documents regarding the subject, I discovered that there are a number of schemes for variable names. We decide to follow the linear algebra scheme.

1 The Problem and the Paradigm

- The input is a list of pairs of points $(\mathbf{a}_1, \mathbf{b}_1)$, $(\mathbf{a}_2, \mathbf{b}_2)$, $\dots$, $(\mathbf{a}_n, \mathbf{b}_n)$ where $\mathbf{a}_i \in \mathbb{R}^\ell$ and $\mathbf{b}_i \in \mathbb{R}^m$. The components of the pairs are supposed to be related by the equation

  $$\mathbf{b}_i = \mathbf{f}(\mathbf{a}_i; \mathbf{x})$$

  where $\mathbf{f}$ is an arbitrary vector function and $\mathbf{x}$ is a vector of unknown parameters.

- The output is a $\mathbf{x}$ that satisfies the above equation for all i.

- Of course, when $(\mathbf{a}_i, \mathbf{b}_i)$ are real data gather from experiments, there will be noise and error. Moreover, we may have more pairs of $(\mathbf{a}_i, \mathbf{b}_i)$ than the degree of freedom of $\mathbf{x}$. As a result, a parameter $\mathbf{x}$ that satisfies equation for all i might not exist.

- Instead, we hope that we can find $\mathbf{x}$ that yields the least error. The **least squares** approach of solving the problem is to choose $\mathbf{x}$ that minimizes the following error:

  $$E_{LS} = \sum_i \|\mathbf{r}_i\|^2 = \sum_i \|\mathbf{b}_i - \mathbf{f}(\mathbf{a}_i; \mathbf{x})\|^2$$

- The vector $\mathbf{r}_i = \mathbf{b}_i - \mathbf{f}(\mathbf{a}_i; \mathbf{x})$ is called the **residual vector**.

2 Linear Least Squares

- An important special case is when $\mathbf{f}(\mathbf{b}; \mathbf{x})$ is linear in $\mathbf{x}$. That is,

  $$\mathbf{f}(\mathbf{a}; \mathbf{x}) = \mathbf{A}(\mathbf{a})\mathbf{x}$$

  where the function $\mathbf{A}(\mathbf{a})$ maps $\mathbf{a}$ to a matrix of size $m \times \ell$.

- The action of f on $\mathbf{a}_1$, $\mathbf{a}_2$, $\ldots$, $\mathbf{a}_n$ can be expressed in matrix form. First, we define

$$\mathsf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{bmatrix}, \text{ and } \mathsf{A} = \begin{bmatrix} \mathbf{A}(\mathbf{a}_1) \\ \vdots \\ \mathbf{A}(\mathbf{a}_n) \end{bmatrix}.$$

  Then, we have that we would like to solve the equation

$$\mathsf{A}\mathbf{x} = \mathsf{b}.$$

- The "linear" least square error is then

$$E_{LS} = E_{LLS} = \|\mathsf{b} - \mathsf{A}\mathbf{x}\|^2.$$

- For further discussion, we define the error vector:

$$\mathsf{r} = \begin{bmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_n \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 - \mathbf{f}(\mathbf{a}_i; \mathbf{x}) \\ \vdots \\ \mathbf{b}_n - \mathbf{f}(\mathbf{a}_n; \mathbf{x}) \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 - \mathbf{A}(\mathbf{a}_1)\mathbf{x} \\ \vdots \\ \mathbf{b}_n - \mathbf{A}(\mathbf{a}_n)\mathbf{x} \end{bmatrix} = \mathsf{b} - \mathsf{A}\mathsf{x}$$

- Linear albegra tells us that the error is minimized when the residual vector is in the null space of A^T. In other words, we solve for r and $\mathbf{x}$ that satisfies the following equations:

$$\begin{aligned} \mathsf{r} + \mathsf{A}\mathbf{x} &= \mathsf{b} \\ \mathsf{A}^T\mathsf{r} &= \mathbf{0} \end{aligned}$$

- Multiplying the top equation by A^T, we have the **normal equation**:

$$\mathsf{A}^T\mathsf{A}\mathbf{x} = \mathsf{A}^T\mathsf{b},$$

  and we can find $\mathbf{x}$ by solving the normal equation.

- The solution to the normal equation is given by

$$\mathbf{x} = (\mathsf{A}^T\mathsf{A})^{-1}\mathsf{A}^T\mathsf{b}.$$

  The matrix $(\mathsf{A}^T\mathsf{A})^{-1}\mathsf{A}^T$ is called the **pseudoinverse** $\mathsf{A}^\dagger$ of A. However, we normally don't find the pseudoinverse to solve the problem.

- A way to solve the normal equation is to compute the Cholesky factorization of $\mathsf{A}^T\mathsf{A}$, which is a symmetric positive definite matrix:

$$\mathsf{A}^T\mathsf{A} = \mathbf{R}^T\mathbf{R}$$

  where $\mathbf{R}$ is upper-triangular. So, we can solve for $\mathbf{x}$ as follows:

  - Solve $\mathbf{R}^T\mathbf{y} = \mathsf{b}$.
  - Solve $\mathbf{R}\mathbf{x} = \mathbf{y}$.

  Solving the normal equation that way can be numerically unstable though.

- Alternatively, we can do QR factorization $\mathsf{A} = \mathbf{QR}$ and notice that

$$E_{LLS} = \|\mathsf{b} - \mathsf{A}\mathbf{x}\|^2 = \|\mathbf{Q}^T\mathsf{b} - \mathbf{Q}^T\mathsf{A}\mathbf{x}\|^2 = \|\mathbf{Q}^T\mathsf{b} - \mathbf{R}\mathbf{x}\|^2$$

  because $\mathbf{Q}$ is orthonomal, thus preserving norms. The normal equation becomes $\mathbf{R}\mathbf{x} = \mathbf{Q}^T\mathsf{b}$, and we can solve for $\mathbf{x}$ in this one instead.

- We would like to point out the form of the linear least square error:

$$\begin{aligned} E_{LLS} &= (\mathsf{b} - \mathsf{A}\mathbf{x})^T(\mathsf{b} - \mathsf{A}\mathbf{x}) \\ &= \mathbf{x}^T\mathsf{A}^T\mathsf{A}^T\mathbf{x} - 2\mathbf{x}^T\mathsf{A}^T\mathsf{b} + \mathsf{b}^T\mathsf{b} \end{aligned}$$

  This means that E_{LLS} is a function as a function of $\mathbf{x}$ is minimized when

$$\mathsf{A}^T\mathsf{A}\mathbf{x} = \mathsf{b}.$$

2.1 Special Case: Linear Regression for Linear Motion Models

- In computation, we often have $\mathbf{a}_i$ be a point in one space and $\mathbf{b}_i$ is a point in another space.

  For example, $\mathbf{a}_i$ might be the pixel position of a 3D point in a photograph taken by one camera, and $\mathbf{b}_i$ is the pixel position of the same 3D point in a photograph taken by one camera.

- The function $\mathbf{f}$ in the above setting is called a **motion model**.

- Many motion models have *linear relationship* between the displacement $\mathbf{d} = \mathbf{b} - \mathbf{a}$ and the unknown parameter $\mathbf{x}$. That is,

$$\mathbf{d} = \mathbf{b} - \mathbf{a} = \mathbf{J}(\mathbf{a})\mathbf{x}.$$

  where $\mathbf{J}(\mathbf{a}) = \partial\mathbf{f}(\mathbf{a};\mathbf{x})/\partial\mathbf{x}$ is the Jacobian of $\mathbf{f}$ with respect to $\mathbf{x}$.

- For example, a similarity transformation in 2D is defined by four parameters $\mathbf{x} = (t_x, t_y, \alpha, \beta)$, and the motion function is defined as

$$\mathbf{f}\left(\begin{bmatrix} x \\ y \end{bmatrix}; t_x, t_y, \alpha, \beta\right) = \begin{bmatrix} 1+\alpha & -\beta & t_x \\ \beta & 1+\alpha & t_y \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + \alpha x - \beta y + t_x \\ y + \beta x + \alpha y + t_y \end{bmatrix} = \begin{bmatrix} \mathbf{f}_x \\ \mathbf{f}_y \end{bmatrix}.$$

  The Jacobian is given by

$$\mathbf{J}(\mathbf{a}) = \begin{bmatrix} \partial\mathbf{f}_x/\partial t_x & \partial\mathbf{f}_x/\partial t_y & \partial\mathbf{f}_x/\partial a & \partial\mathbf{f}_x/\partial b \\ \partial\mathbf{f}_y/\partial t_x & \partial\mathbf{f}_y/\partial t_y & \partial\mathbf{f}_y/\partial a & \partial\mathbf{f}_x/\partial b \end{bmatrix} = \begin{bmatrix} 1 & 0 & x & -y \\ 0 & 1 & y & x \end{bmatrix}$$

  We have that

$$\mathbf{f}(\mathbf{a};\mathbf{x}) - \mathbf{x} = \begin{bmatrix} x + \alpha x - \beta y + t_x \\ y + \beta x + \alpha y + t_y \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \alpha x - \beta y + t_x \\ \beta x + \alpha y + t_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & x & -y \\ 0 & 1 & y & x \end{bmatrix}\begin{bmatrix} t_x \\ t_y \\ \alpha \\ \beta \end{bmatrix} = \mathbf{J}(\mathbf{a})\mathbf{x}.$$

- In this case, we can set up the following **linear regression** problem:

$$\text{minimize } E_{LLS} = \sum_i \|\mathbf{d}_i - \mathbf{J}(\mathbf{a}_i)\mathbf{x}\|^2 = \|\mathsf{d} - \mathsf{J}\mathbf{x}\|$$

  where

$$\mathsf{d} = \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_n \end{bmatrix}, \text{ and } \mathsf{J} = \begin{bmatrix} \mathbf{J}(\mathbf{a}_1) \\ \vdots \\ \mathbf{J}(\mathbf{a}_n) \end{bmatrix}$$

  We can use any of any of the methods for linear least squares to solve for $\mathbf{x}$.

2.2 Weighted Linear Least Squares

- In some situations, we may wish to weight errors of each pairs differently. That is, we might want to minimize

$$E_{WLLS} = \sum_i w_i \|\mathbf{b}_i - \mathbf{A}(\mathbf{a}_i)\mathbf{x}\|^2.$$

  where each w_i is a positive constant.

- We define the weight matrix

$$\mathsf{W} = \begin{bmatrix} \mathrm{diag}(\sqrt{w_1}) & & & \\ & \mathrm{diag}(\sqrt{w_2}) & & \\ & & \ddots & \\ & & & \mathrm{diag}(\sqrt{w_n}) \end{bmatrix}$$

  where $\mathrm{diag}(\sqrt{w_i})$ is the diagonal matrix of size $m \times m$ whose diagonal entries are $\sqrt{w_i}$.

- Using W, we can write the error as

$$E_{WLLS} = \sum_i w_i \|\mathbf{b}_i - \mathbf{A}(\mathbf{a}_i)\mathbf{x}\|^2 = \sum_i \|\sqrt{w_i}\mathbf{b}_i - \sqrt{w_i}\mathbf{A}(\mathbf{a}_i)\mathbf{x}\|^2 = \|\mathsf{W}\mathsf{b} - \mathsf{W}\mathsf{A}\mathbf{x}\|^2$$

- The solution is, again, given by the solution to the normal equation:

$$\mathsf{A}^T\mathsf{W}^T\mathsf{W}\mathsf{A}\mathbf{x} = \mathsf{A}^T\mathsf{W}^T\mathsf{W}\mathsf{b}.$$

2.3 Generalized Linear Least Squares

- In the generalized least square problem, we would like to minimize the error

$$E_{GLLS} = \|\mathsf{V}\mathsf{r}\|^2 = \|\mathsf{V}(\mathsf{b} - \mathsf{A}\mathbf{x})\|^2 = \|\mathsf{V}\mathsf{b} - \mathsf{V}\mathsf{A}\mathbf{x}\|^2$$

  where V is an arbitrary $\mathbb{R}^{mn \times mn}$ matrix.

- The interpretation of the above optimization problem is that we might assume that the residuals of the pairs are correlated instead of being independent. The matrix V is suppose to denote the variance/covariance between the residuals of the pairs.

- The solution then can be found by solving the normal equation:

$$\mathsf{A}^T\mathsf{V}^T\mathsf{V}\mathsf{A}\mathbf{x} = \mathsf{A}^T\mathsf{V}^T\mathsf{V}\mathsf{b}.$$

2.4 Total Linear Least Squares

- In the standard least squares problem, we effectively solve the following optimization problem:

$$\begin{gathered} \text{minimize } \|\mathsf{r}\|_F \\ \text{subjected to } \mathsf{b} + \mathsf{r} \in \mathrm{col}(\mathsf{A}) \end{gathered}$$

  where $\|\mathsf{r}\|_F$ is the Frobenius norm of r, and $\mathrm{col}(\mathsf{A})$ is the column space of A.

  Note that, since r is a vector, we have that $\|\mathsf{r}\|_F = \|\mathsf{r}\|$.

- The interpretation of the above optimization problem is that we find the smallest perturbation r to b so that the equation $\mathsf{A}\mathbf{x} = \mathsf{b} + \mathsf{r}$ has a solution.

- In the total least square problem, we would like to solve the minimization problem:

$$\begin{array}{c}\text{minimize } \left\| \left[\begin{array}{c|c} \mathsf{E} & \mathsf{r} \end{array} \right] \right\|_F \\ \text{subjected to } \mathsf{b} + \mathsf{r} \in \text{col}(\mathsf{A} + \mathsf{E})\end{array}$$

  That is, we allow perturbation to both b and A so that the equation $(\mathsf{A} + \mathsf{E})\mathbf{x} = \mathsf{b} + \mathsf{r}$ has a solution.

- We can generalize the problem a little bit more by solving

$$\begin{array}{c}\text{minimize } \left\| \mathbf{D} \left[\begin{array}{c|c} \mathsf{E} & \mathsf{r} \end{array} \right] \mathbf{T} \right\|_F \\ \text{subjected to } \mathsf{b} + \mathsf{r} \in \text{col}(\mathsf{A} + \mathsf{E})\end{array}$$

  where $\mathbf{D} = \text{diag}(d_1, d_2, \ldots, d_{mn})$ and $\mathbf{T} = \text{diag}(t_1, t_2, \ldots, t_\ell)$ are diagonal matrices of arbitrary scaling factors.

- We rewrite the equation

$$(\mathsf{A} + \mathsf{E})\mathbf{x} = \mathsf{b} + \mathsf{r}$$

  as

$$\left(\mathbf{D} \left[\begin{array}{c|c} \mathsf{A} & \mathsf{b} \end{array} \right] \mathbf{T} + \mathbf{D} \left[\begin{array}{c|c} \mathsf{E} & \mathsf{r} \end{array} \right] \mathbf{T} \right) \mathbf{T}^{-1} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} = \mathbf{0}$$

- The above equation tells us to find $\Delta \in \mathbb{R}^{mn \times (\ell+1)}$ such that $\mathsf{C} + \Delta$ is rank deficient where

$$\mathsf{C} = \mathbf{D} \left[\begin{array}{c|c} \mathsf{A} & \mathsf{b} \end{array} \right] \mathbf{T}.$$

- To do so, we perform the SVD of C:

$$\mathsf{C} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$$

  where $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_{mn}]$, $\mathbf{V} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_{\ell+1}]$, $\mathbf{u}_i \in \mathbb{R}^{mn}$, and $\mathbf{v}_j \in \mathbb{R}^{\ell+1}$. We also have $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\ell+1})$ where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k = \sigma_{k+1} = \cdots = \sigma_{\ell+1}$ are singular values of C.

- The characterization of singular values tells us mininum value of $\|\Delta\|_F$ is $\sigma_{\ell+1}$. The minimum is attained when we set

$$\Delta = -\mathbf{C} \mathbf{v} \mathbf{v}^T$$

  where $\mathbf{v}$ is any unit vecotr in span$\{\mathbf{v}_{k+1}, \mathbf{v}_{k+2}, \ldots, \mathbf{v}_{\ell+1}\}$.

- To find $\mathbf{x}$, we first find a vector $\mathbf{v}$ in the above span having the form

$$\mathbf{v} = \begin{bmatrix} \mathbf{y} \\ \alpha \end{bmatrix}, \alpha \neq 0.$$

  Then,

$$\mathbf{x} = \frac{-1}{\alpha t_{\ell+1}} \mathbf{T}_1 \mathbf{y}, \text{ where } \mathbf{T}_1 = \text{diag}(t_1, t_2, \ldots, t_\ell)$$

- If we cannot find the $\mathbf{x}$ in the last item, then the total least square problem has no solutions.

- When $\sigma_{\ell+1}$ is a repeated singular value, the solution $\mathbf{x}$ is not unique. However, we can easily find a solution $\mathbf{x}_{TLS}$ with the "least norm".

  The process involves finding an orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{mn\times(\ell-k+1)}$ such that

  $$\begin{bmatrix}\mathbf{v}_{k+1} & \mathbf{v}_{k+2} & \cdots & \mathbf{v}_{\ell+1}\end{bmatrix}\mathbf{Q} = \begin{bmatrix}\mathbf{W} & \mathbf{y} \\ \mathbf{0} & \alpha\end{bmatrix} \begin{matrix}\{\ell-k \\ \{1\end{matrix}.$$

  Then, $\mathbf{x}_{TLS} = \frac{-1}{\alpha t_{\ell+1}}\mathbf{T}_1\mathbf{y}$ is the solution such that

  $$\|\mathbf{T}_1^{-1}\mathbf{x}_{TLS}\| \leq \|\mathbf{T}_1^{-1}\mathbf{x}\|$$

  for any solution $\mathbf{x}$.

3 Non-Linear Least Squares

- We might have that $\mathbf{f}$ is a non-linear function of the parameter $\mathbf{x}$. In this case, we have the problem of **non-linear least squares**.
- We solve the non-linear least squares by turning it to a linear least square problem. This is achieved by first order Taylor's series expansion:

  $$\mathbf{f}(\mathbf{a}_i;\mathbf{x}+\Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{a}_i;\mathbf{x}) + \mathbf{J}(\mathbf{a}_i;\mathbf{x})\Delta\mathbf{x}.$$

- The above equation suggests that we use a Newton-like iteration. That is, we first come up with a initial parameter $\mathbf{x}$. Then, we iteratively find update $\Delta\mathbf{x}$ to $\mathbf{x}$ by evaluating

  $$\begin{aligned}
E_{LNS}(\Delta\mathbf{x}) &= \sum_i \|\mathbf{b}_i - \mathbf{f}(\mathbf{a}_i;\mathbf{x}+\Delta\mathbf{x})\|^2 \\
&\approx \sum_i \|\mathbf{b}_i - \mathbf{f}(\mathbf{a}_i;\mathbf{x}) - \mathbf{J}(\mathbf{a}_i;\mathbf{x})\Delta\mathbf{x}\|^2 \\
&= \sum_i \|\mathbf{r}_i - \mathbf{J}(\mathbf{a}_i;\mathbf{x})\Delta\mathbf{x}\|^2 \\
&= \Delta\mathbf{x}^T\left[\sum_i \mathbf{J}^T\mathbf{J}\right]\Delta\mathbf{x} - 2\Delta\mathbf{x}^T\left[\sum_i \mathbf{J}^T\mathbf{r}_i\right] + \sum_i \|\mathbf{r}_i\|^2 \\
&= \Delta\mathbf{x}^T\mathsf{J}^T\mathsf{J}\Delta\mathbf{x} - 2\Delta\mathbf{x}^T\mathsf{J}^T\mathsf{r} + \mathsf{r}^T\mathsf{r}
\end{aligned}$$

  where the definition of r is the same as that in the last section. J is similar to the one in Section 2.1, and the difference is that, now, the Jacobian also depends on $\mathbf{x}$:

  $$\mathsf{J} = \begin{bmatrix}\mathbf{J}(\mathbf{a}_1;\mathbf{x}) \\ \vdots \\ \mathbf{J}(\mathbf{a}_n;\mathbf{x})\end{bmatrix}$$

- We would like to minimize the error. Therefore, we choose $\Delta\mathbf{x}$ so that $E_{LNS}(\Delta\mathbf{x})$ is minimized. To do so, we solve for $\Delta\mathbf{x}$ in the normal equation:

  $$\mathsf{J}^T\mathsf{J}\Delta\mathbf{x} = \mathsf{J}^T\mathsf{r}.$$

  and update $\mathbf{x} + \Delta\mathbf{x}$.
- As with any iterative method, the system can become unstable. We'll talk about two approaches to prevent this.

- The first approach is to choose **step size** α such that $0 < \alpha \leq 1$ and do the update $\mathbf{x} \leftarrow \mathbf{x} + \alpha\Delta\mathbf{x}$ instead.

  A simple way to pick α is to start with 1 and successively halve the value.

- The second approach is to solve for $\Delta\mathbf{x}$ in the equation

$$(\mathsf{J}^T\mathsf{J} + \lambda \mathrm{diag}(\mathsf{J}^T\mathsf{J}))\Delta\mathbf{x} = \mathsf{J}^T\mathsf{r}.$$

  where λ is a damping parameter used to ensure that the system is stable.

- The **Levenberg–Marquardt algorithm** is the combination of the damped Newton-like iteration and the following rules for updating λ:
  - Start with $\lambda = \lambda_0$ and a factor $\nu > 1$.
  - Compute the $E_{LNS}(\Delta\mathbf{x})$ using the current λ and λ/ν.
    - If both of the cases are worst than the current guess, then update $\lambda \leftarrow \lambda\nu$ and repeat the calculation without updating $\mathbf{x}$ until it gets better.
    - If λ/ν results in a reduction in the error, then an update to $\mathbf{x}$ is made and we update $\lambda \leftarrow \lambda/\nu$.
    - If λ/ν results in a worst error, but λ is better, then we make the update to $\mathbf{x}$ with λ staying the same.

  Note that the rule says that, if both λ and λ/ν results in better errors, we always take the λ/ν update.

3.1 Weighted Non-Linear Least Squares

- Just as in the linear least squares case, we might which to weight the error of each sample different. So, we minimize

$$E_{WNLS} = \sum_i w_i \|\mathbf{b}_i - \mathbf{f}(\mathbf{a}_i; \mathbf{x})\|^2$$

- Now, as $\mathbf{f}$ is non-linear, we seek incremental update $\Delta\mathbf{x}$ that minimizes

$$\begin{aligned} E_{WNLS}(\Delta\mathbf{x}) &= \sum_i w_i \|\mathbf{b}_i - \mathbf{f}(\mathbf{a}_i; \mathbf{x} + \Delta\mathbf{x})\|^2 \\ &\approx \sum_i w_i \|\mathbf{r}_i - \mathbf{J}(\mathbf{a}_i; \mathbf{x})\Delta\mathbf{x}\|^2 \\ &= \sum_i \|\sqrt{w_i}\mathbf{r}_i - \sqrt{w_i}\mathbf{J}(\mathbf{a}_i; \mathbf{x})\Delta\mathbf{x}\|^2 \\ &= \|\mathsf{W}\mathsf{r} - \mathsf{W}\mathsf{J}\Delta\mathbf{x}\|^2. \end{aligned}$$

  This tells us to solve the normal equation:

$$\mathsf{J}^T\mathsf{W}^T\mathsf{W}\mathsf{J}\Delta\mathbf{x} = \mathsf{J}^T\mathsf{W}^T\mathsf{W}\mathsf{r}$$

  in each step of the iteration.

- If we use Levenberg-Marquardt, we solve

$$(\mathsf{J}^T\mathsf{W}^T\mathsf{W}\mathsf{J} + \lambda \mathrm{diag}(\mathsf{J}^T\mathsf{W}^T\mathsf{W}\mathsf{J}))\Delta\mathbf{x} = \mathsf{J}^T\mathsf{W}^T\mathsf{W}\mathsf{r}$$

  in each step.

4 Robust Least Squares

- The ordinary least squares that we have been discussed is not robust against the presence of outliers.
- This is a result of the objective function $E_{LS} = \sum_i \|\mathbf{r}_i\|^2$ increases indefinitely with the size of the residuals. As a result, an outlier with large residual can screw up the process.
- Robust approach tries to replace the square of the norm function with a less rapidly increasing **penalty function** $\rho(\|\mathbf{r}\|)$. Its derivative $\psi(\|\mathbf{r}\|) = d\rho(\|\mathbf{r}\|)/d\|\mathbf{r}\|$ is called the **influence function**.
- Using the new penalty function, we now minimize

  $$E_{RLS} = \sum_i \rho(\|\mathbf{r}_i\|) = \sum_i \rho(\|\mathbf{b}_i - \mathbf{f}(\mathbf{a}_i;\mathbf{x})\|)$$

  instead of the sum of squares. The parameter that minimizes the above objective function is called the **M-estimate**.
- To find the parameter $\mathbf{x}$, we solve the equation:

  $$\frac{\partial E_{RLS}}{\partial \mathbf{x}} = \mathbf{0}.$$

  We have that

  $$\begin{aligned}\frac{\partial E_{RLS}}{\partial \mathbf{x}} &= \sum_i \frac{\partial \rho(\|\mathbf{r}_i\|)}{\partial \mathbf{x}} = \sum_i \frac{d\rho(\|\mathbf{r}_i\|)}{d\|\mathbf{r}_i\|}\frac{\partial \|\mathbf{r}_i\|}{\partial \mathbf{x}} = \sum_i \frac{d\rho(\|\mathbf{r}_i\|)}{d\|\mathbf{r}_i\|}\frac{\mathbf{r}_i^T}{\|\mathbf{r}_i\|}\frac{\partial \mathbf{r}_i}{\partial \mathbf{x}} = \sum_i \left(\frac{1}{\|\mathbf{r}_i\|}\frac{d\rho(\|\mathbf{r}_i\|)}{d\|\mathbf{r}_i\|}\right)\mathbf{r}_i^T\frac{\partial \mathbf{r}_i}{\partial \mathbf{x}} \\ &= \sum_i \left(\frac{1}{\|\mathbf{r}_i\|}\frac{d\rho(\|\mathbf{r}_i\|)}{d\|\mathbf{r}_i\|}\right)\frac{\partial \|\mathbf{r}_i\|^2}{\partial \mathbf{x}} \\ &= \sum_i w(\|\mathbf{r}_i\|)\frac{\partial \|\mathbf{r}_i\|^2}{\partial \mathbf{x}}\end{aligned}$$

  where $w(r) = \frac{1}{r}\frac{d\rho(r)}{dr}$ is called the **weight function**.
- Pretending that $w(\|\mathbf{r}_i\|)$ is constant for all i, we have that any $\mathbf{x}$ that satisfies

  $$\sum_i w(\|\mathbf{r}_i\|)\frac{\partial \|\mathbf{r}_i\|^2}{\partial \mathbf{x}} = \mathbf{0}$$

  yields a local minimum of the error function

  $$E_{IRLS} = \sum_i w(\|\mathbf{r}_i\|)\|\mathbf{r}_i\|^2.$$

  Of course, $w(\|\mathbf{r}_i\|)$ depends on $\mathbf{x}$, but the above simplification suggests the following **iteratively reweighted least squares** algorithm:

  - Start with an initial guess $\mathbf{x}^{(0)}$.
  - Loop until convergence.
    - Compute the weight $w_i = w(\|\mathbf{r}^{(k)}\|) = w(\|\mathbf{b}_i - \mathbf{f}(\mathbf{a}_i;\mathbf{x}^{(k)})\|)$ for all i.
    - Solve the weighted least square problem

      $$\text{minimize} \sum_i w_i\|\mathbf{r}_i^{(k+1)}\|^2 = \sum_i w_i\|\mathbf{b}_i - \mathbf{f}(\mathbf{a}_i;\mathbf{x}^{(k+1)})\|^2$$

      to get $\mathbf{x}^{(k+1)}$.

- Here are some popular influence functions:
  - L_2: $\psi(r) = r$.
  - L_1: $\psi(r) = \mathrm{sgn}(r)$.
  - Huber:

$$\psi(r) = \begin{cases} |r|, & |r| < \varepsilon \\ \varepsilon, & |r| \geq \varepsilon \end{cases}$$

  - Tukey's biweight:

$$\psi(r) = \begin{cases} r(1 - r^2/\varepsilon^2)^2, & |r| < \varepsilon \\ 0, & |r| \geq \varepsilon \end{cases}$$

- Note that the L_2 and L_1 norms are not, well, robust. The more robust ones (Huber and Tukey's biweight) include a parameter ε which limits the influence of outliers.
- The ε is supposed to be chosen to be set to the variance of the inliers. However, estimating the variance from the residuals of all the pairs would contaminate the variance with the large residual of the outlier
- An effective way to compute the ε parameter is to use the **median absolute deviation** (MAD):

$$MAD = \mathrm{median}(\|\mathbf{r}_1\|, \|\mathbf{r}_2\|, \ldots, \|\mathbf{r}_n\|).$$

  We typically set $\varepsilon = 1.438 \times MAD$.
- It is unclear to me if we compute ε only once before the first iteration of if we recompute it in every iteration. Literature, though, suggests that ε is problem dependent, which means we know it before we start optimizing.)

5 Back Propagation