Outline

1. Nonlinear Least Squares Problems
   - Introduction
   - Example / Background

2. Special Case: Linear Least Squares
   - Quick Review / Crash Course
In least squares problems, the objective function $f$ has a special form

$$f(\bar{x}) = \frac{1}{2} \sum_{j=1}^{m} r_j(\bar{x})^2, \quad \bar{x} \in \mathbb{R}^n$$

we refer to each $r_j$ as a **residual**. We assume, for now, that $m \geq n$ so that we have more residuals than dimensions (independent variables). [ **OVER-DETERMINED** ]

The least squares formulation is useful for fitting model parameters to data and has applications in a wide range of fields: chemistry, physics, engineering, finance, economics, etc.

It answers the question **“What model (in a certain class) best fits the observed data?”**
The least-squares-objective has a special form, which makes it easier to solve than general non-linear minimization problems:

We assemble the \textbf{residual vector} 

\[
\bar{r}(\bar{x}) = [r_1(\bar{x}), r_2(\bar{x}), \ldots, r_m(\bar{x})]^T.
\]

Hence, the objective can be written as 

\[
f(\bar{x}) = \frac{1}{2} \bar{r}(\bar{x})^T \bar{r}(\bar{x}) = \frac{1}{2} \| \bar{r}(\bar{x}) \|^2.
\]

We are going to express the derivatives of \( f(\bar{x}) \) in terms of the \textbf{Jacobian} of \( \bar{r}(\bar{x}) \), which is the \( m \times n \) matrix of first partial derivatives defined by 

\[
J(\bar{x}) = \left[ \frac{\partial r_j(\bar{x})}{\partial x_i} \right]_{i=1,2,\ldots,n}^{j=1,2,\ldots,m}
\]
With the Jacobian notation we can write

\[
\nabla f(\bar{x}) = \sum_{j=1}^{m} r_j(\bar{x}) \nabla r_j(\bar{x}) = J(\bar{x})^T \bar{r}(\bar{x})
\]

\[
\nabla^2 f(\bar{x}) = \sum_{j=1}^{m} \nabla r_j(\bar{x}) \nabla r_j(\bar{x})^T + \sum_{j=1}^{m} r_j(\bar{x}) \nabla^2 r_j(\bar{x})
\]

\[
= J(\bar{x})^T J(\bar{x}) + \sum_{j=1}^{m} r_j(\bar{x}) \nabla^2 r_j(\bar{x})
\]

Usually \(J(\bar{x})\) can be computed explicitly without too much work. This gives us a way to get the gradient \(\nabla f(\bar{x})\). Further, this gives us the first “half” of the Hessian \(\nabla^2 f(\bar{x})\) for “free,” i.e. without computing any second derivatives.

In many applications, the second part of the Hessian is small. When this happens we can exploit this by approximating \(\nabla^2 f(\bar{x}) \approx J(\bar{x})^T J(\bar{x})\) so that we have a good approximation of the Hessian, without computing any second derivatives!!!
All our previously defined minimization algorithms can be applied to the least squares problem

$$\min_{\bar{x} \in \mathbb{R}^n} f(\bar{x}) = \frac{1}{2} \min_{\bar{x} \in \mathbb{R}^n} \|\bar{r}(\bar{x})\|_2^2$$

In essence, we just take our old algorithms, and change them to exploit the special structure of the gradient and Hessian.

Prior to hammering out all the gory details, lets take a closer look at the origins of nonlinear least-squares problems.
Example: We study the effect of a certain medication on a patient. Blood is drawn at certain times \( \{t_j\} \) after the patient takes a dose — the concentration of the medication in the patient’s blood-stream \( \{y_j\} \) is measured.

We think that the following model is a good description of the process

\[
\Phi(\bar{x}; t) = x_1 + x_2 t + e^{-x_3 t}
\]

Here, \( x_1, x_2, \) and \( x_3 \) are the parameters of the model (to be determined), and \( t \) indicates time.

We seek to determine the parameters so that the discrepancy between the concentrations predicted by the model \( \{\Phi(\bar{x}; t_j)\} \), and the observed concentrations \( \{y_j\} \) are minimized in the least squares sense.
Figure: An illustration of the discrepancy between the model (solid blue line), and the measurements (red dots). The size of the deviation is indicated by the solid red vertical lines.
The least-squares error is measured by the objective

\[ f(\bar{x}) = \frac{1}{2} \sum_{j=1}^{m} \left[ y_j - \Phi(\bar{x}; t_j) \right]^2 \]

Note that at this point \( \{t_j, y_j\}_{j=1}^{m} \) are known, and the values \( \bar{x} \) are unknown.

By solving the least-squares-problem

\[ \bar{x}^* = \arg \min_{\bar{x} \in \mathbb{R}^n} f(\bar{x}) \]

we find the model

\[ \Phi(\bar{x}^*; t_j) = x_1^* + x_2^* t_j + e^{-x_3^* t_j} \]

which best fits the measurements.
Possible model for Ca\textsuperscript{2+} ion concentration in a cardiocyte during the relaxation phase:

\[ c(t) = A e^{-\alpha t} + B e^{-\beta t}. \]

Figure: Neonatal cardiocyte.
Least Squares Fitting: Background

The previous example (\#1) is an instance of what is known as a **fixed-regressor model** in statistics. It assumes that the times \( \{t_j\} \) at which we draw blood are known to high accuracy, while the observations \( \{y_j\} \) contain “random” errors due to equipment limitations and/or human error.

The least-squares objective is by far not the only way to measuring the discrepancy, we could use

\[
\sum_{j=1}^{m} \left[ y_j - \Phi(\bar{x}; t_j) \right]^2, \text{ or } \sum_{j=1}^{m} \left| y_j - \Phi(\bar{x}; t_j) \right|, \text{ or } \max_{j=1,2,\ldots,m} \left| y_j - \Phi(\bar{x}; t_j) \right|
\]

However, the sum-of-squares measure is

(i) easier to work with

(ii) (usually) the correct choice for statistical reasons...
Let $\epsilon_j$ denote the discrepancy at measurement $\#j$, i.e.

$$\epsilon_j = y_j - \Phi(\bar{x}; t_j)$$

In many cases it is reasonable to assume that the $\epsilon_j$ are independent and identically distributed ("iid"), with a variance $\sigma^2$ and probability density function $g_\sigma(\cdot)$.

This assumption will often be true, e.g. when the model accurately reflects the actual process, and when the errors do not contain a "systematic" component.

Under this assumption, the likelihood of a particular set of observations $\{y_j\}$ given that the actual parameter vector is $\bar{x}$ is given by:

$$p(\bar{y}; \bar{x}, \sigma) = \prod_{j=1}^{m} g_\sigma(\epsilon_j)$$
Since the observations \( \{y_j\} \) are known, the most likely value of \( \bar{x} \) is obtained by maximizing \( p(\bar{y}; \bar{x}, \sigma) \) with respect to \( \bar{x} \). The resulting value \( \bar{x}^* \) is called the maximum likelihood estimate of the parameters.

When the discrepancies are assumed to be normally distributed, we have

\[
g_\sigma(\bar{\epsilon}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{\epsilon^2}{2\sigma^2} \right)
\]

so that

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

It is clear that \( p(\bar{y}; \bar{x}, \sigma) \) is maximized when the sum-of-squares \( \sum_{j=1}^{m} [y_j - \Phi(\bar{x}; t_j)]^2 \) is minimized.
Summary (Statistical motivation)
When the discrepancies are assumed to be independent, identically distributed with a normal distribution function, the maximum likelihood estimate is obtained by minimizing the sum of the squares.

These assumptions on \( \{\epsilon_j\} \) are very common, but do not describe the only situation for which the minimizer of the sum-of-squares makes statistical sense.

Disclaimer: With apologies to all real statisticians out there...
When each function $r_j(\bar{x})$ is linear, the Jacobian $J$ is constant, and we have

$$f(\bar{x}) = \frac{1}{2} \| J\bar{x} + \bar{r}_0 \|^2_2, \quad \bar{r}_0 = \bar{r}(0).$$

the gradient and Hessian are also simple expressions

$$\nabla f(\bar{x}) = J^T (J\bar{x} + \bar{r}_0), \quad \nabla^2 f(\bar{x}) = J^T J.$$

The objective is convex; solving for the stationary point $\nabla f(\bar{x}^*) = 0$ gives the system of equations

$$J^T J\bar{x}^* = -J^T \bar{r}_0,$$

this system of equations is known as the **normal equations**.
The linear least squares problem is of interest since many models used in practice $\Phi(\bar{x}; t)$ are linear.

The linear least squares problem is really a question of numerical linear algebra (Math 543, and Math 541), but given its importance it is worth taking a quick look at three algorithms for finding the solution.

We assume:

- $m \geq n$. (O\textsc{ver-d}\textsc{etermined}: More measurements than parameters)
- $J$ has full column rank.

The Cholesky factorization $R^T R = J^T J$ (where $R$ is $n \times n$ upper triangular, and $J$ is $m \times n$) is guaranteed to exist when these assumptions are true.
Approach #1: Direct solution of the Normal Equations.

- Compute the coefficient matrix \( J^T J \) and the right-hand-side \(-J^T \bar{r}_0\).
- Compute the Cholesky factorization \( R^T R = \text{cholesky}(J^T J) \) of the symmetric matrix \( J^T J \).
- Perform a forward and backward substitution with the Cholesky factors to recover the solution \( \bar{x}^* \).

This approach has one significant disadvantage. — The condition number of \( J^T J \)

\[
\text{cond}(J^T J) = \frac{\|\lambda\|_{\text{max}}(J^T J)}{\|\lambda\|_{\text{min}}(J^T J)} = \text{cond}(J)^2 = \left[ \frac{\sigma_{\text{max}}(J)}{\sigma_{\text{min}}(J)} \right]^2
\]

is the square of the condition number of \( J \).
The relative error of the computed solution is (usually) proportional to the condition number, the fact that \( \text{cond}(J^T J) = \text{cond}(J)^2 \) is very bad news indeed when \( J \) is ill-conditioned.

**Note:** \( J^T J \) is essentially a Hilbert matrix.

In the worst case scenario, the Cholesky factorization may break down due to roundoff errors when \( J \) is ill-conditioned!

**Approach #2: QR-factorization of \( J \) — \( J\Pi = QR \), where \( Q \) is orthonormal, and \( R \) upper triangular**

Since the Euclidean norm is invariant under orthogonal transformations, we have

\[
\|J\bar{x} + \bar{r}_0\|_2 = \|U(J\bar{x} + \bar{r}_0)\|_2
\]

for any \( m \times m \) orthogonal matrix \( U \).
Suppose we perform (Math 543) a QR-factorization with column pivoting on the matrix $J$ to obtain

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

where

- $\Pi$ is an $n \times n$ permutation matrix ($\Rightarrow$ orthogonal)
- $Q$ is $m \times m$ orthogonal
- $Q_1$ is the first $n$ columns of $Q$.
- $Q_2$ is the remaining $(m - n)$ columns of $Q$.
- $R$ is $n \times n$ upper triangular
This gives us

\[ \| J\bar{x} + \bar{r}_0 \|_2^2 = \left\| \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} (J \Pi \Pi^T \bar{x} + \bar{r}_0) \right\|_2^2 = \left\| \begin{bmatrix} R \\ 0 \end{bmatrix} (\Pi^T \bar{x}) + \begin{bmatrix} Q_1^T \bar{r}_0 \\ Q_2^T \bar{r}_0 \end{bmatrix} \right\|_2^2 = \| R (\Pi^T \bar{x}) + Q_1^T \bar{r}_0 \|_2^2 + \| Q_2^T \bar{r}_0 \|_2^2 \]

The second part is unaffected by \( \bar{x} \), but setting the first term to zero minimizes \( \| J\bar{x} + \bar{r}_0 \|_2^2 \), i.e. we find

\[ \bar{x}^* = -\Pi R^{-1} Q_1^T \bar{r}_0 \]

In practice, \( R\bar{z} = -Q_1^T \bar{r}_0 \) is solved by backward substitution, and then \( \bar{x}^* = \Pi \bar{z} \).
The QR-based approach does not square the condition number of $J$. The relative error of the solution will be proportional to a value in the range $[\text{cond}(J), \text{cond}(J)^2]$, usually $\ll \text{cond}(J)^2$, rather than $\text{cond}(J)^2$ for the direct solution of the normal equations.

**In most situations, the QR-based approach is the way to go.**

However, if/when we require maximal robustness and/or want to extract more information about the sensitivity of the solution to errors in $J$ or $\bar{r}_0$ we can bring out the big hammer —

**Approach #3: Singular Value Decomposition (SVD) of $J$.**

The SVD [*mathematics*] is known by many names: the Proper Orthogonal Decomposition (POD), the Karhunen-Loève (KL-) Decomposition [*signal analysis*], Principal Component Analysis (PCA) [*statistics*], Empirical Orthogonal Functions, etc...
Table: The many names, faces, and close relatives of the Singular Value Decomposition...
Special Case: Linear Least Squares

Hits on scholar.google.com.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Principal.Component.Analysis</td>
<td>672,000</td>
<td>874,000</td>
<td>1,140,000</td>
<td>1,340,000</td>
</tr>
<tr>
<td>Singular.Value.Decomposition</td>
<td>158,000</td>
<td>178,000</td>
<td>219,000</td>
<td>256,000</td>
</tr>
<tr>
<td>Karhunen.Loeve</td>
<td>21,700</td>
<td>23,700</td>
<td>27,300</td>
<td>29,300</td>
</tr>
<tr>
<td>Canonical.Correlation.Analysis</td>
<td>22,600</td>
<td>25,100</td>
<td>29,200</td>
<td>32,600</td>
</tr>
<tr>
<td>Empirical.Orthogonal.(Function</td>
<td>Functions)</td>
<td>16,800</td>
<td>19,600</td>
<td>22,800</td>
</tr>
<tr>
<td>Proper.Orthogonal.Decomposition</td>
<td>7,850</td>
<td>9,340</td>
<td>12,500</td>
<td>15,200</td>
</tr>
<tr>
<td><strong>11/2016</strong></td>
<td><strong>11/2017</strong></td>
<td><strong>11/20nn</strong></td>
<td><strong>11/20nn</strong></td>
<td></td>
</tr>
<tr>
<td>Principal.Component.Analysis</td>
<td>1,800,000</td>
<td>1,940,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Singular.Value.Decomposition</td>
<td>337,000</td>
<td>407,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Karhunen.Loeve</td>
<td>33,400</td>
<td>38,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canonical.Correlation.Analysis</td>
<td>42,200</td>
<td>49,500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Empirical.Orthogonal.(Function</td>
<td>Functions)</td>
<td>32,400</td>
<td>38,000</td>
<td></td>
</tr>
<tr>
<td>Proper.Orthogonal.Decomposition</td>
<td>18,800</td>
<td>22,400</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table:** The many names, faces, and close relatives of the Singular Value Decomposition...
The SVD of $J$ is given by (Math 543)

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

where

- $U$ is $m \times m$ orthogonal
- $U_1$ contains the first $n$ columns of $U$
- $U_2$ contains the remaining $(m - n)$ columns of $U$
- $V$ is $n \times n$ orthogonal
- $S$ is $n \times n$ diagonal, with elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$.

Note that $J^T J = VS^2 V^T$, so that the columns of $V$ are eigenvectors of $J^T J$ with eigenvalues $\sigma_j^2$. 
Now,\[ \| J\bar{x} + \bar{r}_0 \|_2^2 = \left\| \begin{bmatrix} S \\ 0 \end{bmatrix} (V^T\bar{x}) + \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} \bar{r}_0 \right\|_2^2 \]
\[ = \left\| S(V^T\bar{x}) + U_1^T\bar{r}_0 \right\|_2^2 + \left\| U_2^T\bar{r}_0 \right\|_2^2 \]

Again, we find the optimum by setting the first contribution to zero, i.e.
\[ \bar{x}^* = VS^{-1}U_1^T\bar{r}_0 = \sum_{i=1}^{n} \frac{\bar{u}_i^T\bar{r}_0}{\sigma_i} \bar{v}_i, \]

where \( \bar{u}_i \) and \( \bar{v}_i \) are the \( i \)-th columns of \( U \) and \( V \), respectively.
The expression for the optimum,

$$\bar{x}^* = \sum_{i=1}^{n} \frac{\bar{u}_i^T \bar{r}_0}{\sigma_i} \bar{v}_i$$

gives us information about the sensitivity of $\bar{x}^*$. When $\sigma_i$ is small, $\bar{x}^*$ is particularly sensitive to perturbations that affect $\bar{u}_i^T \bar{r}_0$.

This information is useful when $\sigma_n / \sigma_1 \ll 1$ ($J$ nearly rank-deficient).
Summary: Three Methods for $J^T J \vec{x}^* = -J^T \vec{r}_0$.

All three approaches are useful under the right circumstances

- Cholesky-based algorithm is particularly useful when $m \gg n$, in this case it is practical to store $J^T J$, but not $J$. When $J$ is rank-deficient or ill-conditioned diagonal pivoting must be implemented to limit the propagation of round-off errors. (*This approach to be used sparingly*)

- In the QR-approach with column pivoting, ill-conditioning usually causes the elements in the lower right-hand corner of the matrix $R$ to be much smaller than the other elements. The strategy produces a solution to a nearby problem in which $J$ is slightly perturbed. (*This is the preferred every-day approach*)
• The SVD-approach is the most robust and reliable for ill-conditioned problems. When \( J \) is actually rank deficient, some of the singular values \( \sigma_i \) are exactly zero. Any vector of the form

\[
\bar{x}^* = \sum_{i: \sigma_i \neq 0} \frac{\bar{u}_i^T \bar{r}_0}{\sigma_i} + \sum_{i: \sigma_i = 0} \tau_i \bar{v}_i
\]

(for any values \( \tau_i \)) is a minimizer of the least-squares problem. Usually the minimum-norm (\( \tau_i = 0 \)) solution is desirable. *(When \( J \) is rank-deficient, this is the only approach of the three that works)*

With these results in our tool-box, we are ready to attack the solution of the non-linear least squares problem next time.
Index