Least-Squares Problems

Daniel P. Robinson
Department of Applied Mathematics and Statistics
Johns Hopkins University

October 21, 2020

Outline

The least-squares problem

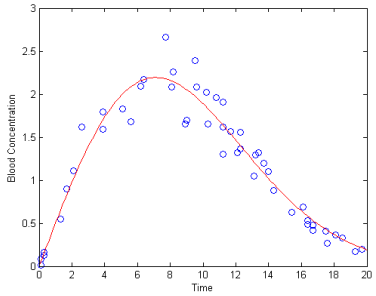
Given a function $F : \mathbb{R}^n \to \mathbb{R}^m$, find a vector x that solves the optimization problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; \tfrac{1}{2}\|F(x)\|_2^2$$

- this is an unconstrained optimization problem with objective function

$$f(x) = \tfrac{1}{2}\|F(x)\|_2^2 = \tfrac{1}{2}\sum_{i=1}^{m} F_i(x)^2$$

- we could blindly apply the linesearch and trust-region methods that we have already developed
- we prefer methods that take advantage of the special structure of f, i.e., it is a squared-sum of other functions
- will assume throughout that F is at least once-continuously differentiable
- the type of least-squares problem is determined by the properties of F
  - if $F(x) = Ax - b$ for some matrix $A \in \mathbb{R}^{m \times n}$ and vector $b \in \mathbb{R}^m$, then it is a linear least-squares problem
  - otherwise, it is a nonlinear least-squares problem



Example 1.1 (Linear and nonlinear least-squares problem)

Given data (t_i, y_i) for $i = 1, 2, \ldots, m$, find the parameter vector $x \in \mathbb{R}^n$ that solves

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; \tfrac{1}{2}\|\Phi(x) - y\|_2^2$$

where

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \quad \text{and} \quad \Phi(x) = \begin{pmatrix} \phi(x; t_1) \\ \phi(x; t_2) \\ \vdots \\ \phi(x; t_m) \end{pmatrix}.$$

- a model of $\phi(x; t) = x_1 t + x_2 e^t$ implies a linear least-squares problem
- a model of $\phi(x; t) = x_1 t + \cos(x_2 t)$ implies a nonlinear least-squares problem

parameter space

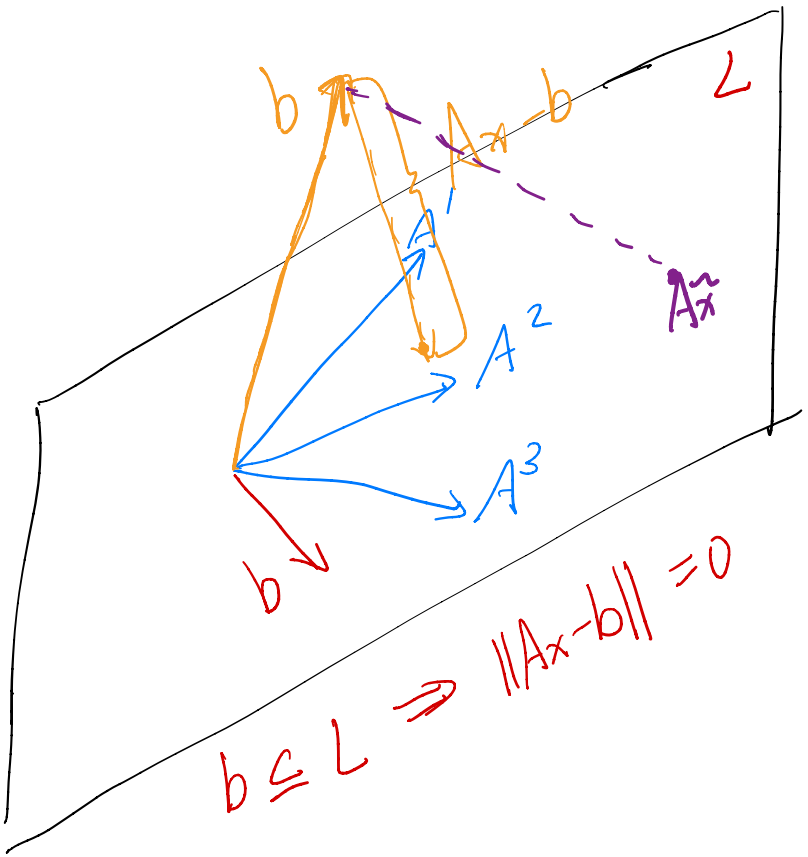
linear least square problem

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax-b\|_2^2 = \frac{1}{2}(Ax-b)^T(Ax-b)$$
$$= \frac{1}{2}x^TA^TAx - b^TAx + \frac{1}{2}b^Tb$$

Assume A has full rank

$$A = [A^1 \; A^2 \; \cdots \; A^n]$$

$$Ax = x_1A^1 + x_2A^2 + \cdots + x_nA^n$$



$L \subseteq \mathbb{R}^m$

$b \subseteq L \Rightarrow \|Ax-b\| = 0$

To find the minimum of $\|Ax-b\|_2$, we need $Ax-b$ is perpandicular to $L \Rightarrow Ax-b$ is perpandicular to each A^i

$$\Rightarrow A^i(Ax-b) = 0 \quad \forall \; i = 1, 2, \cdots n.$$

$$\begin{bmatrix} \cdots A^1 \cdots \\ \cdots A^2 \cdots \\ \cdots A^n \cdots \end{bmatrix} (Ax-b) = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$A^T(Ax-b) = 0$$

$$\Rightarrow A^TAx = A^Tb \longleftrightarrow \text{normal equation}$$

Way to solve this problem

① Cholesky factorization

$\min \frac{1}{2} x^T Q x - c^T x \quad Q \succ 0$

solution: $Qx = c$

$Q = L L^T \rightarrow L$ is lower trienguler matrix

$\hookrightarrow L L^T x = c$

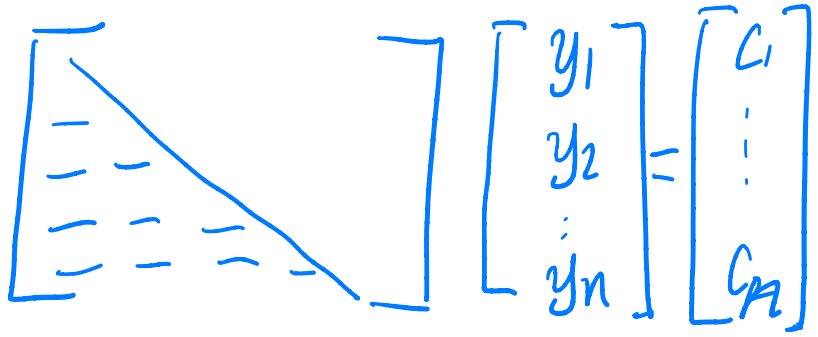
$\underbrace{L^T x}_{y} = c$

$\hookrightarrow Ly = c$

solve by backsubsitiution

$Ly = c$

L



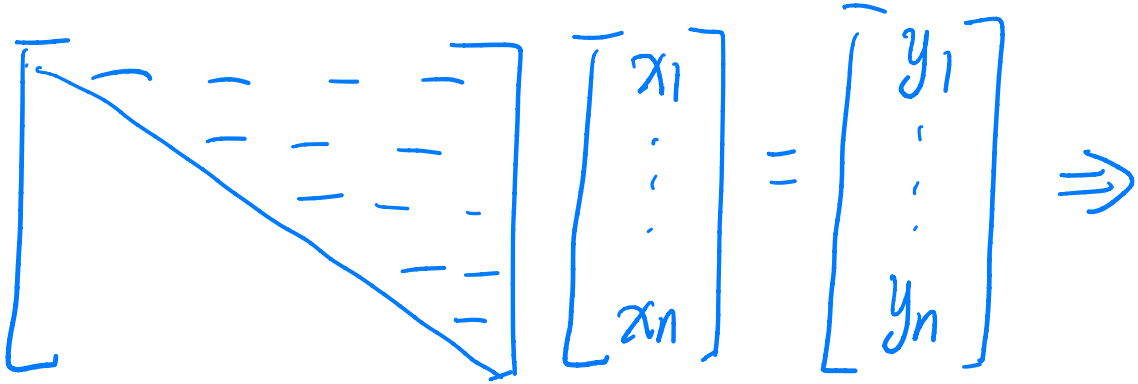
$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}$

$L_{11} \cdot y_1 = c_1 \rightarrow y_1$

$\hookrightarrow L_{21} y_1 + L_{22} y_2 = c_2 \rightarrow y_2$

$\hookrightarrow L_{31} y_1 + L_{32} y_2 + L_{33} y_3 = c_3 \cdots$

$$L^T x = y$$



$$\Rightarrow \begin{cases} L_{nn} x_n = y_n \rightarrow x_n \\ L_{n\,n-1} x_n + L_{n-1\,n-1} x_{n-1} = y_{n-1} \rightarrow x_{n-1} \\ \vdots \\ = y_1 \rightarrow x_1 \end{cases}$$

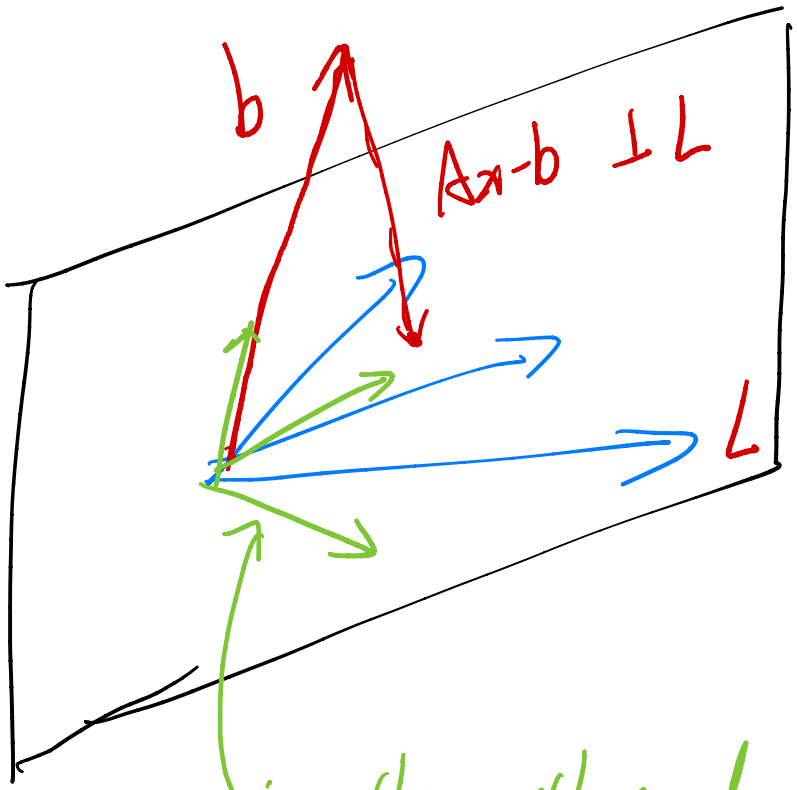
$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2 = \min_{x \in \mathbb{R}^n} \frac{1}{2} x^T \underbrace{A^T A}_{Q} x - x^T \underbrace{A^T b}_{c} + \cdots$$

$Q = A^T A = L L^T$ (cholesky decompostions)

$$L L^T x = A^T b$$

* A is full column rank !!! * numerical stability $\approx (\text{cond}(A))^2$

QR decomposition → GSO



is the orthogonal basis for L

It's easier to compute the projection of b into L with an orthogonal basis rather then A basis.

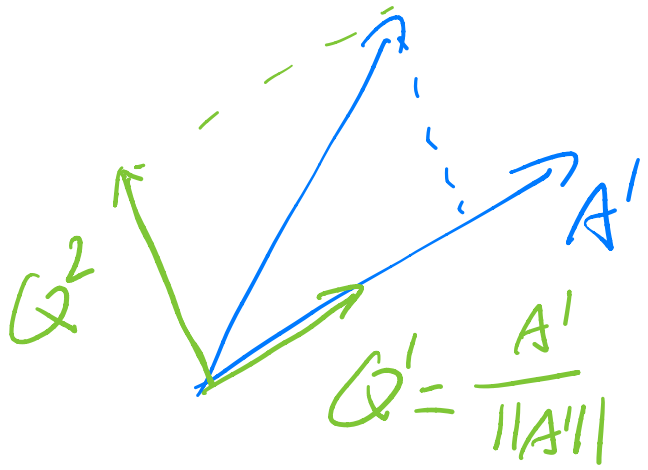
By adding up the projection in orthgonal basis. we get the projection, then we can convert it back to A basis

$A^1, A^2, A^3, \cdots A^n \xrightarrow{GSO} Q^1, \cdots Q^n$

s.t. ① $\|Q^i\| = 1 \quad \forall i = 1, \cdots, n$

② $\langle Q^i, Q^j \rangle = 0 \quad \forall i \neq j$

③ $\text{span}(Q^1, \cdots Q^n) = \text{span}(A^1, \cdots A^n)$



From ③ $\text{span}(Q^1, \cdots Q^n) = \text{span}(A^1, \cdots A^n)$

$A = Q R$

$m \times n \quad m \times n \quad n \times n$

$$\underbrace{\begin{bmatrix} A^1 A^2 \cdots A^n \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} Q^1 Q^2 \cdots Q^n \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1n} \\ 0 & R_{22} & \cdots & R_{2n} \\ \vdots & 0 & R_{33} & \vdots \\ 0 & 0 & & R_{nn} \end{bmatrix}}_{R}$$

① Q has orthogonal column

② R is upper-triangular matrix

Projection of b on L

$$\hookrightarrow \langle b, Q^1\rangle Q^1 + \langle b, Q^2\rangle Q^2 + \cdots + \langle b, Q^n\rangle Q^n$$

$$= (b^T Q^1) Q^1 + \cdots + (b^T Q^n) Q^n$$

$$= Q Q^T b$$

Need to express this projection in the basis A

$$\boxed{Ax = QQ^T b}$$

$$QRx = QQ^T b$$

Q is full rank

Q is positive definite matrix

$$Rx = Q^T b$$ → solve by back-substitution

upper triangular matrix

* numerical stability is better then L

SVD approach

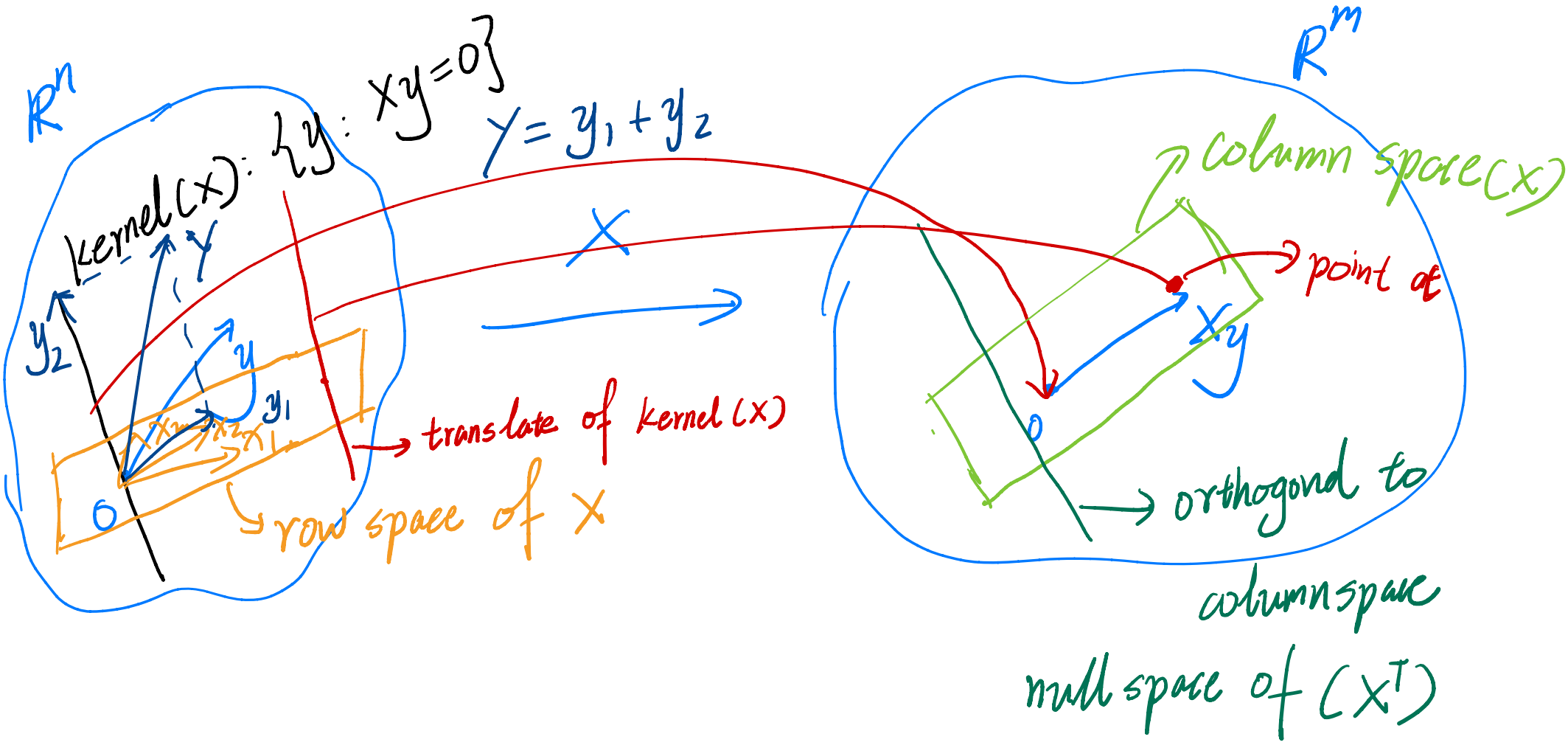
Efficiency perspective: most expensive

Reliability perspective: numerical stable ✓

$X \subseteq \mathbb{R}^{m \times n}$

$X : \mathbb{R}^n \longrightarrow \mathbb{R}^m$

$y \longrightarrow Xy$



$$X = \begin{bmatrix} — & x_1 & — \\ — & x_2 & — \\ & \vdots & \\ — & x_m & — \end{bmatrix} \begin{bmatrix} y \end{bmatrix} = 0$$

↘ kernel of X

each row basis is mapped into column space (X)

↘ column space at R^m

Any vector y can be represented as $y = y_1 + y_2$

y_1: kernel (X)

y_2: row spaces (X)

$$Xy = X(y_1 + y_2) = \underbrace{Xy_1}_{\to 0} + \underbrace{Xy_2}_{\text{map to column space}}$$

$$X = \begin{bmatrix} | & | & & | \\ x^1 & x^2 & \cdots & x^n \\ | & | & & | \end{bmatrix}$$

dim (rowspace) = dim (columnspace)

Singular value decomposition (SVD) is to find such orthgonal basis for these two space in term of X

orthogonal basis of rowspace (X) → orthgonal basis of columnspace (X)

$v^1 \; v^2 \cdots v^m \longrightarrow u^1 \cdots u^n$

$X v^i = \sigma_i u^i$

$\sigma_i \geq 0$

each basis is map into the specific basis in corresponding space

SVD Theorem: $\exists$ orthogonal basis $v^1, v^2, \dots v^q$ for rowspace (X) and

$\exists$ orthogonal basis $u_1, u_2, \dots u_q$ for the columnspace (X) and $\sigma_1 \dots \sigma_q > 0$

S.T. $Xv_i = \sigma_i u_i \quad \forall \; i = 1, \dots q$

$\sigma_1, \sigma_2, \dots \sigma_q$ called singular value

Given $y \in \mathbb{R}^m$: ① need to project y onto row space $\rightarrow \tilde{y}$

② Express $\tilde{y}$ in the orthogonal basis $v^1, v^2 \dots v^q$

③ scale these projection with corresponing singular value σ_i and express in the basis $u_1 \dots u_q$

$$\tilde{y} \xrightarrow{① + ②} \langle y, v^1\rangle v^1 + \langle y, v^2\rangle v^2 + \cdots + \langle y, v^\ell\rangle v^\ell$$

$$= \sum_{i=1}^{\ell} \langle y, v^i\rangle v^i$$

$$\downarrow \quad X v^i = \sigma_i u^i$$

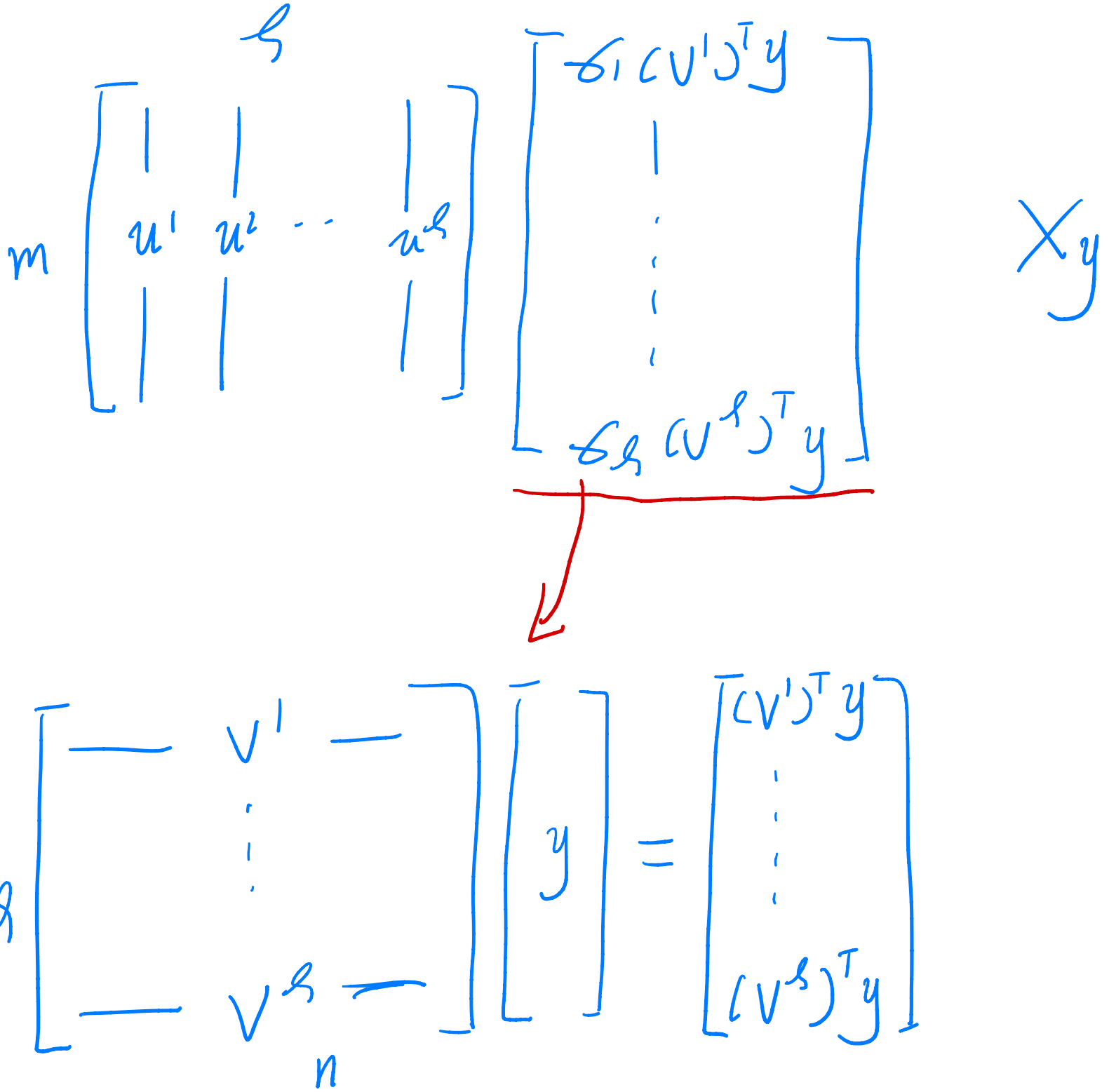
$$Xy = X(\hat{y} + \tilde{y}) = X\hat{y} + X\tilde{y} = \sum_{i=1}^{\ell} \langle y, v^i\rangle X v^i$$

($X\hat{y} = 0$)

$$= \sum_{i=1}^{\ell} \langle y, v^i\rangle \sigma_i u^i$$

Xy expressed in orthgonal basis u_i

$$Xy = \sum_{i=1}^{\ell} \sigma_i \langle y, v^i\rangle u^i \qquad \langle y, v^i\rangle = (v^i)^T y$$


k
m
u^1 u^2 ⋯ u^k
$\sigma_1 (v^1)^T y$
$\sigma_k (v^k)^T y$
X_y
k
v^1
v^k
n
y
=
$(v^1)^T y$
$(v^k)^T y$

$$\begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_k \end{bmatrix} \begin{bmatrix} (v_1)^T y \\ \vdots \\ (v^k)^T y \end{bmatrix} = \begin{bmatrix} \sigma_1 (v_1)^T y \\ \vdots \\ \sigma_k (v^k)^T y \end{bmatrix}$$

Put these together

$$m \begin{bmatrix} | & | & & | \\ u^1 & u^2 & \cdots & u^k \\ | & | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_k \end{bmatrix} k \begin{bmatrix} — & v^1 & — \\ — & v^2 & — \\ & \vdots & \\ — & v^k & — \end{bmatrix} \begin{bmatrix} y \end{bmatrix}$$

k over the middle matrix; n over the v matrix

$k \times y$

$$U = \begin{bmatrix} | & | & & | \\ u^1 & u^2 & \cdots & u^s \\ | & | & & | \end{bmatrix} \qquad V = \begin{bmatrix} | & | & & | \\ v^1 & v^2 & \cdots & v^s \\ | & | & & | \end{bmatrix}$$

$X = \tilde{U} \tilde{\Sigma} \tilde{V}^T \rightarrow$ compact SVD

$V = [v^1, v^2, \ldots v^s, \underbrace{v^{s+1} \ldots v^n}]$

↓

span kernel (X)

So that $v^1, v^2, \ldots v^s$ are orthogonormed.

$$U = [u^1, u^2, \dots u^g, u^{g+1}, \dots u^m]$$

$u^{g+1}, \dots u^m$ → span kernel (X^T)

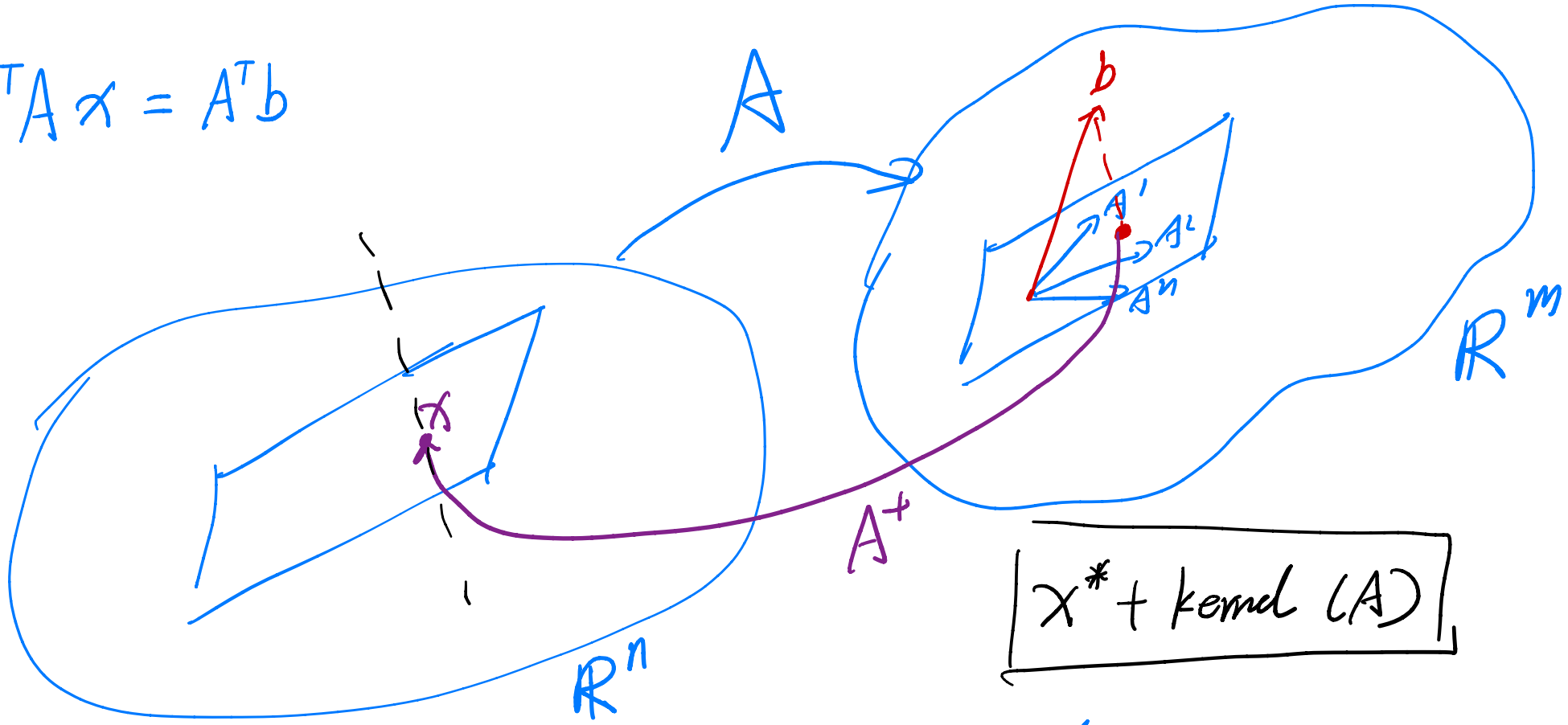
$$\Sigma = \begin{bmatrix} \sigma_1 & & & 0 \\ & \ddots & & \\ & & \sigma_g & \\ 0 & & & 0 \\ & & & \ddots \\ & & & 0 \end{bmatrix} \quad (g \text{ rows})$$

$$X = U \Sigma V^T$$

$$X^\dagger = \hat{V} \hat{\Sigma}^{-1} \hat{U}^T$$

$$\min \frac{1}{2}\|Ax-b\|_2^2$$

$$A^TAx = A^Tb$$



The solution to the linear least square problem is simply

$$A^+b \Rightarrow x = \tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^Tb.$$

$$= V\Sigma^{-1}U^Tb$$

The linear least-squares problem

Given a matrix $A \in \mathbb{R}^{m \times n}$ and vector $b \in \mathbb{R}^m$, find a vector x that solves the optimization problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; f(x) = \tfrac{1}{2}\|Ax - b\|_2^2$$ → convex

- will assume that $m \geq n$
- since $f(x) = \frac{1}{2}b^Tb - x^TA^Tb + \frac{1}{2}x^TA^TAx$, it follows that

$$\nabla f(x) = A^TAx - A^Tb \quad \text{and} \quad \nabla^2 f(x) = A^TA$$

- f is convex since $\nabla^2 f(x) = A^TA \succeq 0$
- if A has full column rank, then f is strictly convex since $\nabla^2 f(x) = A^TA \succ 0$
- if x satisfies

$$A^TAx = A^Tb \qquad \text{(the normal equations)}$$

then x is a solution to the least-squares problem! (Why?)

Solving the normal equations using a Cholesky factorization

1. Form the matrix A^TA and vector A^Tb.
2. Compute a Cholesky factorization $R^TR = A^TA$, where the nonsingular matrix R is upper-triangular with positive diagonal elements.
3. Compute y^* as the solution of the linear system

$$R^Ty = A^Tb$$

4. Compute the linear least-squares solution x^* as the solution to the linear system

$$Rx = y^*$$

- Cholesky factorization is only guaranteed to exist if A has full column rank
- x^* satisfies

$$A^TAx^* = A^Tb$$

  - the accuracy of x^* depends on the condition number of A^TA which is the square of the condition number of A
  - $\mathbf{cond}(A) = \sigma_{\max}/\sigma_{\min}$
  - preferable to have the least-squares solution depend on the condition number of A!
- particularly attractive if $m \gg n$ and n is not too large so that the Cholesky factorization is efficient

If we compute the QR factorization

$$A\Pi = Q\begin{pmatrix} R \\ 0 \end{pmatrix}$$

where

- Π : is an $n \times n$ permutation matrix
- Q : is an $m \times m$ orthogonal matrix
- Q_1 : is the first n columns of Q
- Q_2 : is the last $m - n$ columns of Q
- R : is an $n \times n$ upper-triangular matrix with positive diagonal elements

it then follows that

$$\begin{aligned}
\|Ax - b\|_2^2 &= \|Q^T(Ax - b)\|_2^2 = \|Q^T(A\Pi\Pi^Tx - b)\|_2^2 \\
&= \|Q^TA\Pi\Pi^Tx - Q^Tb\|_2^2 \\
&= \left\| \begin{pmatrix} R \\ 0 \end{pmatrix} \Pi^Tx - \begin{pmatrix} Q_1^Tb \\ Q_2^Tb \end{pmatrix} \right\|_2^2 \\
&= \left\| \begin{pmatrix} R\Pi^Tx - Q_1^Tb \\ -Q_2^Tb \end{pmatrix} \right\|_2^2
\end{aligned}$$

Thus, the least-length solution x^* that minimizes $\|Ax - b\|_2^2$ is the unique solution to

$$R\Pi^Tx = Q_1^Tb$$

Solving the linear least-squares problem using a QR factorization

1. Compute the factorization

$$A\Pi = Q\begin{pmatrix} R \\ 0 \end{pmatrix} = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix}\begin{pmatrix} R \\ 0 \end{pmatrix}$$

2. Form Q_1^Tb.
3. Compute y^* as the unique solution to

$$Ry = Q_1^Tb.$$

4. Compute the least-length solution x^* as the solution to

$$\Pi^Tx = y^*$$

i.e., obtain x^* by permuting entries of y^* according to the permutation matrix Π.

- matrix R being nonsingular relies on A having full column rank
- error in the computed x^* is usually of the same magnitude as the condition number of A (contrast this with the Cholesky factorization approach)

If we compute the SVD of A given by

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

where

- U : is an $m \times m$ orthogonal matrix
- U_1 : is the first n columns of U
- U_2 : is the last $m - n$ columns of U
- V : is an $n \times n$ orthogonal matrix
- S : is an $n \times n$ diagonal matrix with diagonal elements $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_n > 0$

it then follows that

$$\begin{aligned} \|Ax - b\|_2^2 &= \|U^T(Ax - b)\|_2^2 = \|U^T A^T x - U^T b\|_2^2 \\ &= \left\| \begin{pmatrix} S \\ 0 \end{pmatrix} V^T x - \begin{pmatrix} U_1^T b \\ U_2^T b \end{pmatrix} \right\|_2^2 \\ &= \left\| \begin{pmatrix} SV^T x - U_1^T b \\ -U_2^T b \end{pmatrix} \right\|_2^2 \end{aligned}$$

Thus, the least-length solution x^* that minimizes $\|Ax - b\|_2^2$ is the unique solution to

$$SV^T x = U_1^T b \quad \Longrightarrow \quad x^* = VS^{-1}U_1^T b$$

Solving the linear least-squares problem using the SVD

1. Compute the SVD
$$A = U \begin{pmatrix} S \\ 0 \end{pmatrix} V^T = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} S \\ 0 \end{pmatrix} V^T$$
2. Form $y = U_1^T b$.
3. Form $z = S^{-1}y$ by scaling the elements of y by the inverse of the diagonal elements of S
4. Define the least-length solution x^* as
$$x^* = Vz$$

- the SVD is very expensive to compute
- the SVD is the most stable and reliable
- x^* as defined above satisfies
$$x^* = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i$$
- if A does not have full column rank, the least-squares solution of least-length is
$$x^* = \sum_{i:\sigma_i > 0} \frac{u_i^T b}{\sigma_i} v_i$$

Computing an approximate least-squares solution using linear CG

If A has full column rank, then $A^T A \succ 0$ so that the linear conjugate gradient (CG) method may be used to (approximately) solve the normal equations

$$A^T Ax = A^T b$$

- the linear CG algorithm may be used with
$$\text{“}A\text{“} = A^T A \quad \text{and} \quad \text{“}b\text{“} = A^T b$$
- allows for an approximate solution of the linear least-squares problem
- linear CG only requires a single matrix-vector product of the form $A^T Av$ per iteration
  - first, compute product $y = Av$
  - second, compute required product $A^T Av = A^T y$
- may be used for problems with very large m and/or n provided A is sparse
- an iterative method by Paige and Saunders called LSQR is very similar to CG, but deals with A directly instead of $A^T A$.

The nonlinear least-squares problem

Given a function $F : \mathbb{R}^n \to \mathbb{R}^m$, find a vector x that solves the optimization problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; f(x) = \tfrac{1}{2}\|F(x)\|_2^2$$

- will assume that $m \geq n$
- if F is sufficiently differentiable, we have
$$\nabla f(x) = J(x)^T F(x) \quad \text{and} \quad \nabla^2 f(x) = J(x)^T J(x) + \sum_{i=1}^{m} \nabla^2 F_i(x) F_i(x)$$
where J is the Jacobian matrix of F given by
$$J(x) := \nabla F(x) = \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdots & \frac{\partial F_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \frac{\partial F_m}{\partial x_2} & \cdots & \frac{\partial F_m}{\partial x_n} \end{pmatrix}$$ $m \times n$
- f is now typically nonconvex
- f is convex if $J(x)^T J(x) + \sum_{i=1}^{m} \nabla^2 F_i(x) F_i(x)$ is positive semi-definite for all x
- x is a first-order solution of the nonlinear least-squares problem if it satisfies
$$\nabla f(x) = J(x)^T F(x) = 0$$
so we search for these points!

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x) = \tfrac{1}{2}\|F(x)\|_2^2$$

- Could charge ahead using the linesearch and trust-region methods from earlier.
- For linesearch, we would solve subproblems of the form

$$\underset{p\in\mathbb{R}^n}{\text{minimize}}\ f(x) + \nabla f(x)^T p + \tfrac{1}{2}p^T B p$$

  for some chosen $B \succ 0$ that approximates $\nabla^2 f(x)$
- For trust-region, we could solve subproblems of the form

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\ f(x) + \nabla f(x)^T s + \tfrac{1}{2}s^T B s \quad \text{subject to} \quad \|s\|_2 \le \delta$$

  for some chosen B (not necessarily positive definite) that approximates $\nabla^2 f(x)$
- We will cover specialized linesearch and trust-region methods that take advantage of the special structure of f since it is a sum of squares.

The Gauss-Newton subproblem (perspective 1)

The Gauss-Newton subproblem is given by

$$\underset{p\in\mathbb{R}^n}{\text{minimize}}\ f(x) + p^T\nabla f(x) + \tfrac{1}{2}p^T B p = \tfrac{1}{2}\|F(x)\|_2^2 + p^T J(x)^T F(x) + \tfrac{1}{2}p^T J(x)^T J(x) p$$

- a minimizer p of the Gauss-Newton subproblem satisfies

$$J(x)^T J(x) p = -J(x)^T F(x)$$

$Bp = -g$

- these are just the normal equations for the linear least squares problem

$$\underset{p\in\mathbb{R}^n}{\text{minimize}}\ \tfrac{1}{2}\|J(x)p + F(x)\|_2^2$$

The Gauss-Newton subproblem (perspective 2)

An equivalent formulation of the Gauss-Newton subproblem is to compute p as a minimizer of

$$\underset{p\in\mathbb{R}^n}{\text{minimize}}\ \tfrac{1}{2}\|F(x) + J(x)p\|_2^2$$

- can solve this problem using any of our methods from the previous section.
- we solve the nonlinear least-squares problem by solving a sequence of linear least-squares problems (with a linesearch)

Goal: develop a linesearch method for solving the nonlinear least-squares problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x) = \tfrac{1}{2}\|F(x)\|_2^2$$

Requirement: find a positive-definite matrix B that approximates

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{i=1}^{m} \nabla^2 F_i(x) F_i(x)$$

Observation 1: If J has full column rank, then $B = J(x)^T J(x) \succ 0$

Observation 2: If for all $i = 1, 2, \ldots m$ we have that

$$F_i(x) \approx 0 \quad \text{and/or} \quad \nabla^2 F_i(x) \approx 0$$

then B is a very good approximation to $\nabla^2 f(x)$.

- B is a good approximation if either the ith residual is zero or the ith function F_i is nearly linear near the current solution
- this is true for many applications, but certainly not every application

Advantages of the Gauss-Newton subproblem

$$\underset{p\in\mathbb{R}^n}{\text{minimize}}\ \tfrac{1}{2}\|F(x)\|_2^2 + p^T J(x)^T F(x) + \tfrac{1}{2}p^T J(x)^T J(x) p$$

over the Newton subproblem

$$\underset{p\in\mathbb{R}^n}{\text{minimize}}\ \tfrac{1}{2}\|F(x)\|_2^2 + p^T J(x)^T F(x) + \tfrac{1}{2}p^T \left[J(x)^T J(x) + \sum_{i=1}^{m} \nabla^2 F_i(x) F_i(x) \right] p$$

include the following:

- Newton matrix may be indefinite and thus would need to be modified. Standard modified Newton modifications may be used, but they do not take the problem structure into account.
- Gauss-Newton does not require the second-order derivatives $\nabla^2 F_i(x)$ so it is usually much cheaper to implement.
- There are many applications where $J(x)^T J(x)$ dominates the definition of $\nabla^2 f(x)$ and, therefore, the Gauss-Newton approximation often converges very quickly.
- Gauss-Newton subproblem is equivalent to a linear least-squares problem, which may be solved in many ways (some more efficient/robust than others)
- If $J(x)$ has full column rank and $J(x)^T F(x) \neq 0$, then the Gauss Newton step p_G is a descent direction for f since

$$p_G^T \nabla f(x) = p_G^T J(x)^T F(x) = -p_G^T J(x)^T J(x) p_G = -\|J(x)p_G\|_2^2 < 0$$

Why is the last inequality strict?

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x) = \tfrac{1}{2}\|F(x)\|_2^2$$

Algorithm 1 The Gauss-Newton Method for nonlinear least-squares

1: Input an initial guess x_0.
2: Set $k \leftarrow 0$.
3: **loop**
4: Compute a search direction p_k as the solution to

$$\underset{p\in\mathbb{R}^n}{\text{minimize}}\ \tfrac{1}{2}\|F(x_k) + J(x_k)p\|_2^2$$

SVD
QR
Cholesky

5: Perform a (weak) Wolfe linesearch along p_k to obtain a step length $\alpha_k > 0$.
6: Set $x_{k+1} \leftarrow x_k + \alpha_k p_k$
7: Set $k \leftarrow k + 1$
8: **end loop**

- A termination test such as $\|J(x_k)^T F(x_k)\|_2 \leq \tau \|J(x_0)^T F(x_0)\|_2$ should be included for some stopping tolerance $\tau > 0$.
- A linesearch that satisfies the strong Wolfe conditions is fine.
- A backtracking-Armijo linesearch may also be used.

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x) = \tfrac{1}{2}\|F(x)\|_2^2 \tag{1}$$

Theorem 3.1 (Global convergence result for the Gauss-Newton Method)

Suppose that

1. *the level set $\mathcal{L}$ is bounded, where*
$$\mathcal{L} = \{x : f(x) \leq f(x_0)\}$$
2. *F_i is Lipschitz continuously differentiable in a neighborhood $\mathcal{N}$ of the level set $\mathcal{L}$*
3. *the Jacobian matrix J satisfies the uniform-rank condition, i.e., there exists a $\gamma > 0$ such that*
$$\|J(x)z\|_2 \geq \gamma\|z\|_2$$
for all $x \in \mathcal{N}$ and all z.

Then, it follows that the iterates generated by the Gauss-Newton linesearch method stated as Algorithm 1 for solving the nonlinear least-squares problem (1) *satisfy*

$$\lim_{k\to\infty} J(x_k)^T F(x_k) = 0$$

Proof: Apply the appropriate convergence result from the slides on “Line Search Methods".

Summary of the Gauss-Newton Method for solving nonlinear least-squares problems

- it is linesearch method that utilizes a very special positive-definite matrix to define the quadratic model.
- if the residual $F_i(x_k)$ is close to zero or F_i is nearly linear for all $i = 1, 2, \ldots, m$, then the positive-definite approximation is a very accurate model and can lead to fast superlinear convergence.
- the subproblem is equivalent to a linear least-squares problem.
- the subproblem may be approximately solved using techniques that we have previous discussed such as the linear CG method.
- approximate solves using CG is particularly attractive if m and n are very large and $J(x_k)$ is sparse.
- special care should be used when $J(x_k)$ has dependent columns since then the second-order matrix $J(x_k)^T J(x_k)$ in the quadratic model is only positive semi-definite.
- the issues concerning dependent columns is partly overcome by a trust-region based method, as we now see!

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x) = \tfrac{1}{2}\|F(x)\|_2^2$$

The Levenberg-Marquardt Method is a trust-region method that solves the nonlinear least-squares problem by using the subproblem

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\ m_k(s) \quad \text{subject to} \quad \|s\|_2 \leq \delta_k$$

- $\delta_k > 0$ is the trust-region radius
- the model m_k is the same as for the Gauss-Newton Method, i.e.,
$$m_k(s) = \tfrac{1}{2}\|F(x_k) + J(x_k)s\|_2^2 = \tfrac{1}{2}\|F(x_k)\|_2^2 + s^T J(x_k)^T F(x_k) + \tfrac{1}{2}s^T J(x_k)^T J(x_k)s$$
- consequently, similar local convergence results will hold
  - fast superlinear convergence if either the ith residual is zero or F_i is nearly linear near the local solution for all $i = 1, 2, \ldots, m$.

We know from trust-region theory, that the step s_* is the global solution of the Levenberg-Marquardt subproblem

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\ \ m_k(s) = \tfrac{1}{2}\|F(x_k) + J(x_k)s\|_2^2 \ \ \text{subject to} \ \ \|s\|_2 \le \delta_k$$

iff $\|s_*\|_2 \le \delta_k$ and there exists λ_* such that

1. $\lambda_* \ge 0$
2. $[J(x_k)^T J(x_k) + \lambda_* I]s_* = -J(x_k)^T F(x_k)$
3. $\lambda_*(\|s_*\|_2 - \delta_k) = 0$

Note: $J(x_k)^T J(x_k) + \lambda_* I \succeq 0$ for all $\lambda \ge 0$ since $J(x_k)^T J(x_k) \succeq 0$

Observation: The normal equations associated with the linear least-squares problem

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\ \ \tfrac{1}{2}\left\|\begin{pmatrix} J \\ \sqrt{\lambda} I \end{pmatrix} s + \begin{pmatrix} F \\ 0 \end{pmatrix}\right\|_2^2 \tag{2}$$

are precisely

$$(J^T J + \lambda I)s = -J^T F \tag{3}$$

- The secular equation approach for finding the "exact" solution of the trust-region problem requires solving systems of the form (3).
- Systems of the form (3) may be solved by solving the linear least squares problem (2).
- Additional computational gains are possible (See Nocedal and Wright, "Implementation of a Levenberg-Marquardt Method", page 259.)

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ \ f(x) = \tfrac{1}{2}\|F(x)\|_2^2$$

Algorithm 2 The Levenberg-Marquardt Method for nonlinear least-squares

1: Input an initial guess x_0.
2: Choose $\delta_0 > 0$, $0 < \gamma_d < 1 < \gamma_i$, and $0 < \eta_s \le \eta_{vs} < 1$.
3: Set $k \leftarrow 0$.
4: **loop**
5: Compute a trial step s_k as the solution to

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\ \ m_k(s) = \tfrac{1}{2}\|F(x_k) + J(x_k)s\|_2^2 \ \ \text{subject to} \ \ \|s\|_2 \le \delta_k$$

6: Set $\rho_k \leftarrow [f(x_k) - f(x_k + s_k)] / \Delta m_k(s_k)$.
7: **if** $\rho_k \ge \eta_{vs}$ **then**
8: Set $x_{k+1} \leftarrow x_k + s_k$ and $\delta_{k+1} \leftarrow \gamma_i \delta_k$.
9: **else if** $\rho_k \ge \eta_s$ **then**
10: Set $x_{k+1} \leftarrow x_k + s_k$ and $\delta_{k+1} \leftarrow \delta_k$.
11: **else**
12: Set $x_{k+1} \leftarrow x_k$ and $\delta_{k+1} \leftarrow \gamma_d \delta_k$.
13: **end if**
14: Set $k \leftarrow k + 1$
15: **end loop**

- A termination test such as $\|J(x_k)^T F(x_k)\|_2 \le \tau \|J(x_0)^T F(x_0)\|_2$ should be included for some stopping tolerance $\tau > 0$.

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ \ f(x) = \tfrac{1}{2}\|F(x)\|_2^2 \tag{4}$$

Theorem 3.2 (Global convergence of the Levenberg-Marquardt Method)

Suppose that

1. *the level set $\mathcal{L}$ is bounded, where*
$$\mathcal{L} = \{x : f(x) \le f(x_0)\}$$
2. *F_i is Lipschitz continuously differentiable in a neighborhood $\mathcal{N}$ of the level set $\mathcal{L}$*
3. *each trial step s_k satisfies the Cauchy condition*

$$\Delta m_k(s_k) \ge \tfrac{1}{2}\|J(x_k)^T F(x_k)\|_2 \min\left(\delta_k, \frac{\|J(x_k)^T F(x_k)\|_2}{1 + \|J(x_k)^T J(x_k)\|_2}\right)$$

Then, it follows that the iterates generated by the Levenberg-Marquardt trust-region method stated as Algorithm 2 for solving the nonlinear least-squares problem (4) *satisfy*

$$\lim_{k\to\infty} J(x_k)^T F(x_k) = 0$$

Proof: Apply Theorem 1.10 from slides on "Trust Region Methods".

Summary of the Levenberg-Marquardt Method for solving nonlinear least-squares problems

- it is trust-region method that utilizes a very special positive-definite matrix to define the quadratic model (same as for the Gauss-Newton Method).
- if the residual $F_i(x_k)$ is close to zero or F_i is nearly linear for all $i = 1, 2, \ldots, m$, then the positive-definite approximation is a very accurate model and can lead to fast superlinear convergence.
- the linear system that must be solved when finding a zero of the secular equation associated with computing the "exact" solution of the trust-region subproblem, is equivalent to a special linear least-squares problem.
- the subproblem may be approximately solved using techniques that we have previous discussed such as the truncated linear CG method (Steihaug).
- computing approximate solutions using truncated CG is a particularly attractive option if m and n are very large and $J(x_k)$ is sparse.
- as is often the case with general trust-region methods, the Levenberg-Marquardt Method typically performs better than the Gauss-Newton Method when $J(x_k)$ has dependent columns