1 18th of October 2018 — F. Poloni

1.1 Least squares problem

Fact 1.1. *Given $A \in \mathcal{M}(m, n, \mathbb{R})$ the following conditions are equivalent:*

- *$A^T A$ is positive definite;*
- *A has full column rank;*
- *the columns of A are linear indipendent;*
- *$Ker(A) = \{0\}$.*

Fact 1.2. *Given $A \in \mathcal{M}(m, n, \mathbb{R})$, if $A^T A$ is **positive semidefinite** $f(x) = x^T Q x - q^T x + b^T b$ is **strongly** (or **strictly**) **convex**. In other words, $f(x)$ has a **unique** minimum.*

We find the minimum by solving $\nabla f(x) = 0$,
where $f(x) = x^T A^T A x - 2b^T A x + b^T b$
$\nabla f(x) = 2A^T A x - 2A^T b$ How should we compute this gradient?
We know that $f(x + h) = f(x) + (\nabla f(x))^T h + o(\|h\|)$

$$
\begin{aligned}
f(x+h) &= (x+h)^T A^T A (x+h) - 2b^T A(x+h) + b^T b \\
&= \mathbf{x^T A^T A x} + \mathbf{x^T A^T A h} + \mathbf{h^T A^T A x} + \mathbf{h^T A^T A h} \mathbf{- 2b^T A x} - 2b^T A h + \mathbf{b^T b} \\
&= \mathbf{f(x)} + (\mathbf{2x^T A^T A h} - 2b^T A h) + \mathbf{o(\|h\|)} \\
&= f(x) + (\mathbf{A^T A x - 2A^T b})^{\mathbf{T}} \mathbf{h} + o(\|h\|)
\end{aligned}
\tag{1.1}
$$

So, $\mathbf{\nabla f(x) = A^T A x - 2A^T b}$
We would like to know when the gradient is 0.
$\nabla f(x) \stackrel{?}{=} 0 \Leftrightarrow A^T A x = A^T b$
Since $A^T A$ is a **square** matrix and also **non singular** (which means invertible) we may find x by solving a linear system $x = (A^T A)^{-1} (A^T b)$ via:

- Gaussian elimination;
- LU factorization;
- QR factorization;
- Cholesky factorization (specialyzed method for positive definite matrices) Idea: $A^T A$ can be written as $A^T A = R^T R$, where R is a square, upper triangular matrix.

Why do we need factorization method? Let's compute complexity:

- $A^T A \ \rightarrow \ 2mn^2$, where $m > n$;
- $A^T b \ \rightarrow \ 2mn$;
- Solving $A^T Ax = A^T b$ with gaussian elimination has a computational complexity of $\frac{2}{3}n^3$;
- Cholesky factorization $A^T A = R^T R$ which has a cost of $\frac{1}{3}n^3$

1.1.1 Method of normal equations

This method solves least squares problem and takes his name from the fact that "normal" means orthogonal.

The key idea is using symmetry to skip half of the entries of $A^T A$

If $Ax = b$ can't be solved, since A is tall and thin, we can multiply on both sides by A^T and try again, since the matrix is square now.

The residual $Ax - b$ is orthogonal to any vector $v \in span(A) : (Av)^T (Ax - b) = 0$

Why?

$v^T (A^T Ax - A^T b) = 0$, see Figure 1.1.

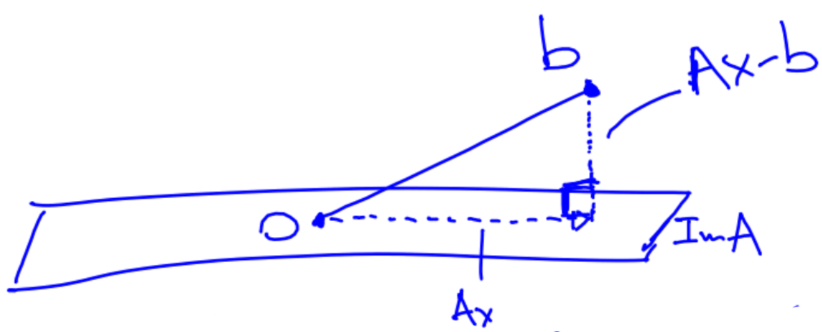

FIGURE 1.1: Geometric idea.

It's possible to find a close formula for solving this problem: $\min\|Ax - b\|$ is given by $x = (A^T A)^{-1} A^T b$.

Definition 1.1 (Moore-Penrose pseudoinverse)**.** *Let A be a matrix in $\mathcal{M}(n, m, \mathbb{R})$, the **Moore-Penrose pseudoinverse** of A with full column rank is* $\mathbf{A}^+ := (A^T A)^{-1} A^T$

So we can write $x = A^+ b$ for the solution of a LS problem.

In particular, the solution of $\|\min Ax - (b_1 + b_2)\|$ is the sum of two solutions of $\min\|Ax_1 - b_1\|$ and $\min\|Ax_2 - b_2\|$.

Obs: $AA^+ = A(A^T A)^{-1} A^T = AA^{-1} A^{T^{-1}} A^T = I_{n \times m}$, but this doesn't hold for $A^+ A = (A^T A)^{-1} (A^T A)$, which is not the identity matrix.

1.2 QR factorization

Theorem 1.3. *$\forall A \in \mathcal{M}(m, m, \mathbb{R})$, $\exists Q \in \mathcal{O}(m, m, \mathbb{R})$ (space of orthogonal matrices of size $m \times m$), $\exists R \in \mathcal{M}(m, m, \mathbb{R})$ upper triangular such that $A = QR$*

QR factorization isn't as powerful as SVD factorization.
Why are we interested in studying factorizations?

- They reveal properties: singularity, rank, . . . ;
- They may be an intermediate step in algorithms.

Example 1.1. *We would like to solve $Ax = b$, with $A \in \mathcal{M}(m, m, \mathbb{R})$ we may:*

1. *first compute the QR factorization ($A = QR$) and then obtain $x = A^{-1}b = R^{-1}Q^{-1}b$*
2. *compute then $c = Q^T b$*
3. *and then $x = R^{-1}c$*

What's the computational cost?

1. $QR \;\rightarrow\; O(m^3)$
2. compute c $\rightarrow\; O(m^2)$
3. compute x $\rightarrow\; O(m^2)$

Let's analyze the case in which A is a vector. Given $x \in \mathbb{R}^m$, we want to find an orthogonal matrix Q such that Qx has the form $\begin{pmatrix} s \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix} = se_1$, where $s = \pm\|x\|$.

We denote e_i the i-th column of the identity matrix.

Definition 1.2 (Housholder reflector)**.** *Let U be a vector in $\mathbb{R}^m$. An* ***Householder reflector*** *is a matrix H such that $H = I - \frac{2}{U^T U} \cdot UU^T$.*

We can observe that $U^T U$ is a scalar.

Since $U^T U = \|U\|^2$, another way of seeing H may be $H = I - \frac{2}{\|U\|^2}UU^T = I - 2vv^T$, where $v = \frac{1}{\|U\|}U$

Lemma 1.4. *Matrices of this form are orthogonal.*

Proof.

$$
\begin{aligned}
HH^T &= (I - \frac{2}{\|U\|^2} \cdot UU^T) \cdot (I - \frac{2}{\|U\|^2} \cdot UU^T) \\
&= I \cdot I - \frac{2}{\|U\|^2} \cdot UU^T I - I \cdot \frac{2}{\|U\|^2} \cdot UU^T + \frac{2}{\|U\|^2} \cdot UU^T \cdot \frac{2}{\|U\|^2} \cdot UU^T \\
&= I - \frac{2}{\|U\|^2} \cdot UU^T - \frac{2}{\|U\|^2} \cdot UU^T + \frac{4}{\|U\|^4} \cdot UU^T UU^T \\
&= I - \frac{4}{\|U\|^2} UU^T + \frac{4}{\|U\|^4} U\|U\|^2 U^T \\
&= I
\end{aligned} \tag{1.2}
$$

□

Example 1.2. $Hx = (I - \frac{2}{\|U\|^2} UU^T)x = x - \frac{2}{\|U\|^2} U(U^T U)$

$y = compute\_product(U, x)$

$a = U' * x$

$b = U' * U$

$y = x \frac{2*a}{b} \cdot U$

All these operations are linear operations, so the complexity is $O(m)$, cheaper than generic matrix-vector product ($O(m^2)$).

Can we find a matrix H in this family that maps x to y (equivalently $Hx = y$)? The answer is given by the following lemma

Lemma 1.5. *$\forall x, y$ s.t $\|x\| = \|y\| \exists H$ s.t. $Hx = y$, choosing $u = x - y$.*

A geometric idea is given by the following

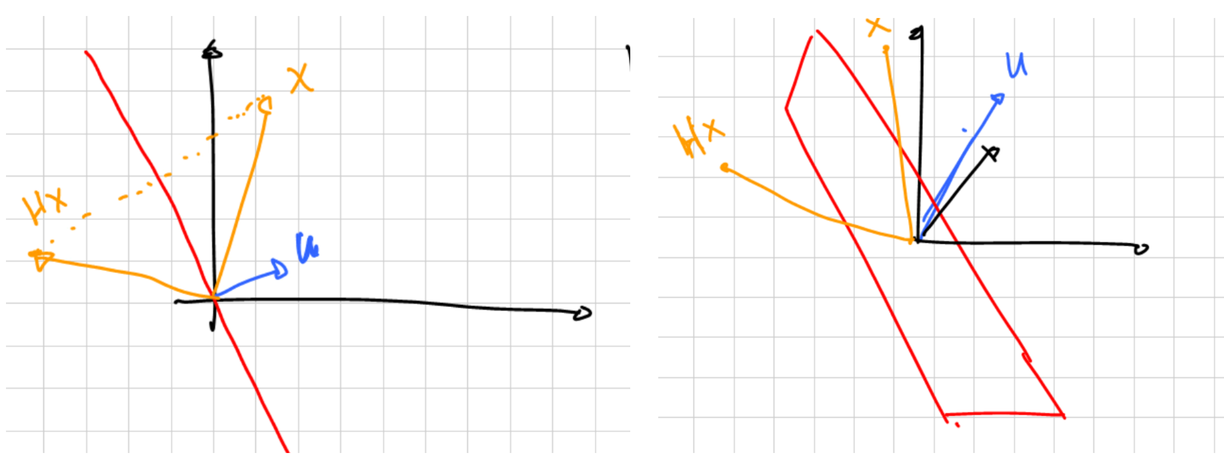

FIGURE 1.2: On the left in $\mathbb{R}^2$, on the right in $\mathbb{R}^3$.

What happens if $y = \begin{pmatrix} \|x\| \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}$?

$y = H^T x$, actually $H^T = H^{-1} = H$

Let's map x to y:

$$U = x - y = \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{pmatrix} - \begin{pmatrix} s \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix} = \begin{pmatrix} x_1 - s \\ x_2 \\ . \\ . \\ . \\ x_n \end{pmatrix}$$

where $s = \|x\|$

Something on Matlab . . .

`function[v,s] = householder_vector(x)`, where v and s are the returned values and x is the argument.